1994年12月1日発行（毎月1回1日発行）第13巻12号通巻152号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

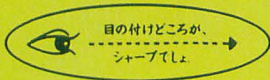
特別付録5"2HD **XL/Imageお試し版＋α**

XL/Image用モデリングツール/各種ゲーム/SX-BASIC最新版など

Oh!X7周年特別企画　懐ゲー制作工房

新製品紹介 H.A.R.P/XDTP SX-68K

12
1994

SOFT BANK　オー／エックス
特別定価900円

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

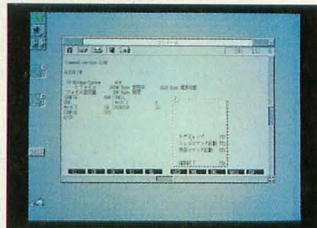
GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境

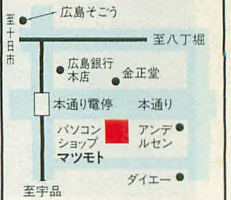
今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

「SX-WINDOW ver3.1システムキット」CZ-296SS（130㎜FD）/CZ-296SSC（90㎜FD） 標準価格22,800円（税別）

X680x0 パワーアップフェア

日時
12月17日(土)・18日(日)
AM11時〜

会場
パソコンショップマツモト
広島市中区本通7-29
TEL(082)504-5785

★うわさの「040 Turboボード」をはじめとした各種周辺機器でパワーアップされたX68の世界を体験して頂きます。

EXEディスク2プレゼント

●官製ハガキに住所、氏名、EXE会員番号と90㎜(3.5型)/120㎜(5.25型)の種別を明記の上、お申込み下さい。また、これからEXEクラブへ入会される方は、商品同梱のEXEクラブ入会申込書に「EXEディスク2希望」と明記の上、ご投函下さい。

応募/問合せ先

〒545 大阪市阿倍野区長池町22-22 シャープ株式会社電子機器事業本部システム機器営業部EXEクラブ事務局EXEディスク2係宛
(TEL 06-621-1221大代表)

申込締切

平成6年12月末日消印有効

●Z'sSTAFF、書体倶楽部はツァイト社の商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。●PostScriptはアドビシステムズ社の登録商標です。●EGWordは株式会社エルゴソフトの登録商標です。

シャープ株式会社

資料請求券 ペリフェラル Oh!X 12月

Oh!X

CONT

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　豊浦史子　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　司馬　護　清瀬栄介　石上達也　柴田　淳　瀧　康史　横内威至　進藤慶到　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら

特別企画　XL/Imageお試し版＋α

Oh!X 7周年特別企画　懐ゲー制作工房

魔法大作戦

新製品紹介　H.A.R.P

新製品紹介　XDTP SX-68K

(で)のショートプロぱーてぃ

表紙絵：塚田　哲也

E　N　T　S

1994 DEC. 12

■広告目次

For
X68030/
X68000series
APPLICATION
SOFTWARE

◎パーソナルDTPをX68で

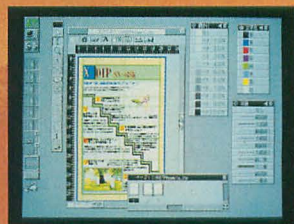
DTP SX-68K　CZ-291BWD　標準価格35,000円(税別)

NEW

縦書きをはじめとした多彩なレイアウト機能で
パーソナルなデスクトップパブリッシングを実現するソフトです。
やさしい操作、豊富な編集機能、グラフィックウィンドウ対応、SX-WINDOWをすでに
ご利用になっている方なら、基本操作を新たに覚えることなく手軽にレイアウトが作成できます。

■豊富なテキスト編集機能：フォント種類、サイズ、文字種の変更はもちろん、上線、下線、網掛け、文字間隔の指定が文字ごとに設定可能。禁則、行間隔、タブ、インデント、マージンもパラグラフ(リターンコードまでの文字列)ごとに設定できます。また各テキストフレームごとに、フレーム形状、リンク状態(テキストの流し込み)、縦書き/横書き、回り込みの設定が可能。検索/置換も単純な文字列だけでなく、スタイル別に行うことができます。
■グラフィックウィンドウに対応：GRW.Xにも対応していますので、いろいろな形状でレイアウトしたグラフィックフレームのデータを65,536色の画像で確認しながらレイアウトできます。
■さまざまな画像フォーマットに対応：ビデオマネージャが対応している静止画フォーマットの他に、「PrintShop PRO-68K」、「CANVAS PRO-68K」、「GScriptファイル」の読み込みに対応しています。

●グラフィックフレーム、テキストフレームとは別に直線、矩形、楕円、多角形が作成できる独立した罫線機能●第1水準を収めた明朝体、ゴシック体のベジェーフォントファイルを標準装備●ページの移動や作成/削除がスピーディに行える独立したページウィンドウをサポート●ページプリンタドライバ(ESC/Page、LIPSIII)を付属、高解像度の美しい印字が可能。またSX-WINDOWが対応しているプリンタも使用可能。

※5MB以上の空きのあるハードディスクが必要です。 (4MB、Ver.3.0)

◎グラフィック感覚の楽譜入力をサポート

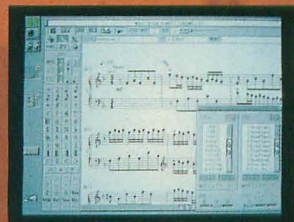
MUSIC SX-68K　CZ-274MWD　標準価格38,000円(税別)

NEW

MIDI、FM、ADPCMに対応した楽譜ワープロ&作曲演奏ソフトです。
自由なレイアウトでグラフィックを描くように楽譜入力、
全パートの同時入力や編集、自動伴奏機能、応用範囲を広げるデータ互換性。
多彩なプリンタ対応で美しい印刷も可能です。

■MIDI、FM、ADPCM対応：MIDI、FM、ADPCMを同時に発音できます。全ての音源を利用した場合、最大発音数は25まで設定可能です。
■全パートの同時入力：ピアノ譜、メロディ譜などの組み合わせで最大16パートまで編集可能。特定パートごとではなく全パートを画面に表示して編集できますので、直接画面上で曲の構成を考えながら作編曲できます。
■コード&リズムによる自動伴奏機能：メロディ上にコードネームとリズムパターンを入力するだけで、自動的に伴奏をつけることができます。
■優れたデータ互換性：「MUSIC PRO-68K」、「MUSIC PRO-68K[MIDI]」のデータファイルが利用できる他、OPM、MML、ZMSファイル形式でデータ出力が可能です。
■多彩なプリンタ対応：ページプリンタドライバ(ESC/Page、LIPSIII)を付属、高解像度の美しい印刷が可能です。
またSX-WINDOWが対応しているプリンタも利用できます。

(4MB、Ver.3.0)

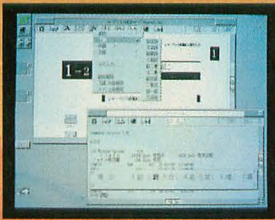
X68030
32bit PERSONAL WORKSTATION

その先のシーンへ。

●さらに実用的なウィンドウシステムへの進化

SX-WINDOW ver3.1 システムキット

CZ-296SS(130㎜FD)/CZ-296SSC(90㎜FD) 標準価格22,800円(税別)

ASK68K Ver3.0を利用したインライン入力のサポート、Human68k/BASICコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できるコンソールのサポートをはじめ、シャーペン.Xをワープロとして利用できるよう機能アップ。また、さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ドローデータ(FSX)/フォントデータ(IFM)処理の高速化も実現しています。

※コンソールでは、SX-WINDOWと処理が重複するものは実行できません。 4MB

●定評のGUI対応ウィンドウワープロ

EGWord SX-68K

CZ-271BWD 標準価格59,800円(税別)

ウィンドウワープロとして評価の高いEGWordのSX-WINDOW対応版。キャラクタベースのワープロを超えたグラフィカルユーザーインターフェイス(GUI)による手軽なDTPソフトとしても優れた表現力を発揮します。定評ある日本語入力方式(EGConvert)によるインライン入力、さまざまなグラフィックデータ(GScript)やテキストデータの貼り込み、また文書互換を実現するEDF(Extended Document Format)形式をサポートしています。 4MB、ver.2.0

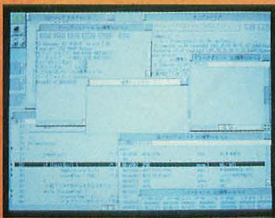
※5MB以上の空きのあるハードディスクが必要です。

●SX-WINDOW開発支援ツール

SX-WINDOW 開発キット Workroom SX-68K

CZ-288LWD 標準価格39,800円(税別)

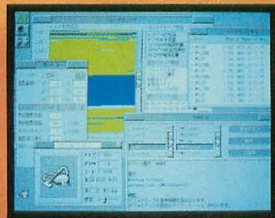
SX-WINDOW用のソフト開発に必要なツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解が深まる33種(基礎編23種、応用編4種、実用編6種)のサンプルプログラム付き。

※ご使用に当ってはC compiler PRO-68K ver.2.1が必要です。

4MB、ver.2.0

●SX-WINDOW開発キットのサポートツール

開発キット用ツール集

CZ-289TWD 標準価格12,800円(税別)

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールの簡易リファレンスを簡単に検索するインサイドSX、イベントの発生を常時監視・確認するイベントハンドラ、リアルタイムにメモリブロックの利用状況を表示するヒープビューアなど11種のツールが用意されています。

4MB、ver.2.0

●SX-WINDOW対応ドローイングツール

Easydraw SX-68K

CZ-264GWD 標準価格19,800円(税別)

イラスト、フローチャート、地図、見取り図など各種グラフィックが製図感覚で作成できます。作成したデータは他のSX-WINDOW対応アプリケーションでも利用でき、企画書などの作成をサポート。ページプリンタドライバも標準装備。 4MB、ver.3.0

●ウィンドウ対応グラフィックツール

Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。 2MB、ver.1.1

●SX-WINDOWを楽しく使うためのアクセサリ集

SX-WINDOW デスクアクセサリ集

CZ-290TWD 標準価格14,800円(税別)

SX-WINDOWをさらに便利に楽しく使うためのデスクアクセサリ集です。スクリーンセーバ、スクラップブック、スケジューラ、アドレス帳、電子手帳通信ツール、パズルなど、12種の豊富なアクセサリが収められています。 4MB、ver.3.0

●マルチタスク機能をはじめ通信環境がさらに充実

Communication SX-68K

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションを実行中でも簡単に通信が可能。自動ログイン機能やプログラム機能、など豊富な機能をサポートしています。 2MB、ver.1.1

●FM音源サウンドエディタ

SOUND SX-68K

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成、変更できるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。 2MB、ver.1.1

●SX-WINDOW対応になってさらにパワーアップ

倉庫番リベンジ SX-68K ユーザー逆襲編

CZ-293AW(130㎜FD)/CZ-293AWC(90㎜FD) 各標準価格6,800円(税別)

倉庫番10年にわたるユーザーの投稿など、新作306面が目白押し。まさに倉庫番の最強版がSX-WINDOW上で楽しめます。AI機能やエディット機能、キャラクタ変更機能も装備。半年で解けたらあなたは天才?です。 2MB、ver.1.1

シリーズ

●X68030/X68000対応

C COMPILER PRO-68K ver.2.1 NEW KIT

CZ-295LSD 標準価格44,800円(税別)

※メインメモリ2MB以上が必要です。

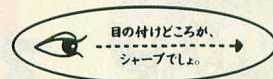
C compiler PRO-68KのX68030/X68000対応版。MPU68030、MC68882の命令セットに対応したアセンブラ、デバッガ、ソースコードデバッガを付属。またHuman68k ver.3.0、ASK68K ver.3.0にも対応。新たにGPIBライブラリ、MC68882対応フローとライブラリを付属しています。

※ 2MB,ver.1.1 の表示は、メインメモリ2MB以上、SX-WINDOW ver.1.1以上が必要であることを示します。

●EGWord、EGConvertは株式会社エルゴソフトの登録商標です。●ESC/Pageはセイコーエプソン株式会社の登録商標です。

●お問い合わせは…シャープ㈱電子機器事業本部(液映)システム機器推進プロジェクトチーム 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ シャープ株式会社

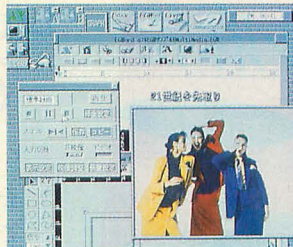
資料請求券 CZソフト Oh!X 12係

ビデオグラフィックスの
世界へ。
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
資料請求券
X68030

1,677万色対応、ビデオ映像を高画質・高速取り込み

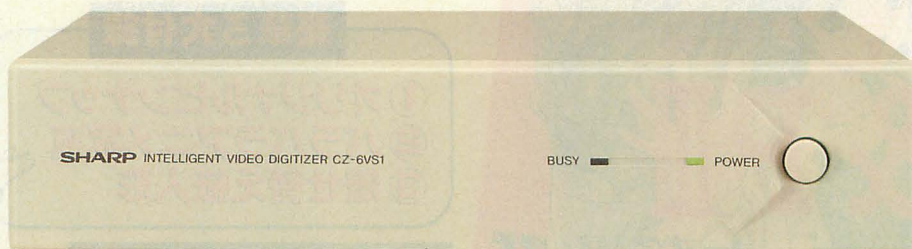
テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば"ビデオスキャナ"とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

※SX-WINDOW版はバージョン3.0以降(メモリー4MB以上)、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上(メモリー8MB以上)が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

SHARP INTELLIGENT VIDEO DIGITIZER CZ-6VS1　BUSY　POWER

■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2(FAST)インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2(FAST)対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

ビデオ入力ユニット

CZ-6VS1

標準価格178,000円(税別)

MIKI

スーパーリアル麻雀 PII&PIII ファンブック

KASUMI

しょう子・香澄・未来……
あの3人にまた会える!!

「……むかし、むかしあるところに麻雀のたいそう強い娘が3人いた。その娘たちの名前は、しょう子、香澄、未来。それはそれはかわいくて愛くるしい娘たちだった。多くの人々が、100円玉を握りしめ、その娘たちに会いに行ったという。しかし、彼女たちの本当の姿を目にした者は少なかった。そして、いつしか人々は娘たちを"麻雀の女神"と呼ぶようになった……」

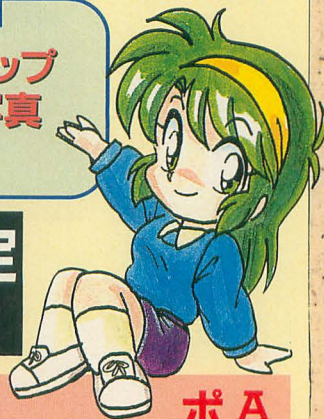
(スーパーリアル麻雀神話・序章より抜粋)

今や伝説・神話となった名作PⅡPⅢ初の公式ファンブック。幻の原画・コンテや描き下ろしセル画、コミックなど貴重な資料が満載の完全保存版!

SHOWKO

豪華3大付録

① オリジナルピンナップ
② パラパラアニメ写真
③ 着せ替え紙人形

PⅡ&PⅢ 初の公式ブック登場!!

11月下旬発売予定
定価2,000円

スーパーリアル麻雀PⅣ 原画&設定資料集

- 定価2,000円
- A4判

豪華ゲスト描き下ろしイラスト満載

A3描き下ろしポスターつき

■定価は税込みです

ソフトバンク株式会社/出版事業部 販売局 TEL 03-5642-8101

すべてのゲームファンに贈る!

ザ・ナムコ・グラフィティ1

完全保存版!

往年のゲームサウンド収録のCD付録つき

NG総集編＆特別編集号

あの『NG』が今、よみがえる!

※写真は『NG』創刊号です。

株式会社ナムコ 監修

「パックマン」「ラリーX」「ディグダグ」「マッピー」「ゼビウス」など、誰もが一度は遊んだことのある名作ゲームを生みだし、今も「リッジレーサー」などの傑作を世に送り出し続けるナムコ。このナムコが発行しゲームファンに絶大な支持を得ていた広報誌『NG』の歴史を中心に、遊びをクリエイトし続けるナムコの姿と歴史を豊富な資料と取材により紹介。往年の名作ゲーム8作品（初収録作品あり）のゲームサウンドを収録したオリジナルCD付録つきです。

定価2,300円（税込）

11月下旬発売予定!

お近くの書店でお求めください

ソフトバンク株式会社/出版事業部

SOFT BANK

GAME BEST SELECTION

「ペンドラゴン」リプレイ 三つの槍の探索

佐藤俊之 著／健部伸明 監修　　定価1,800円

テーブルトークRPG専門誌で好評を博した連載リプレイ「三つの槍の探索」。プレイに使用されたゲームシステム「ペンドラゴン」は、アーサー王伝説をテーマにしたRPGで、ゲームデザイナーはグレッグ・スタフォード、カバーおよび本文のイラストにはマンガ家のたいらはじめを起用。連載時のファンはもちろん、新しいRPGプレイヤーやアーサー王伝説に興味のある読者にも楽しめます。

蓬莱学園108の謎

ゆうせぶん 著／柳川房彦 監修　　定価1,500円

南海の孤島、宇津帆島にある私立高校、蓬莱学園は、生徒総数10万人、教師数1800人、140の公認クラブと、数千を数える同好会が存在する世界最大の高校です。この高校を舞台にした物語は、ネットゲームからはじまり、テーブルトークRPG、小説、ドラマCDとその世界を広げてきました。本書は、蓬莱学園のさまざまな事柄についてQ&A方式で、わかりやすく解説します。

好 評 発 売 中 !

「ファー・ローズ・トゥー・ロード」リプレイ RPGセッションガイド

遊演体 監修　司史生/ゆうせぶん 著　定価1,600円

国産テーブルトークRPGシリーズの最新作、「Far Roads to Lord」の製作にあたったスタッフによる、初の公式ガイドブック。リプレイを中心に、ルールのリファレンスや、背景世界ユルセルームの解説を盛り込み、「F·ローズ」のマスターおよびプレイヤーに、その魅力とプレイ方法を紹介しています。

SIMCITY2000 パーフェクトガイド

中島理彦 著　定価1,600円

大人気シミュレーションゲーム「SIMCITY2000」の完全攻略ガイド。地下の高度利用モード、上下水道、地下鉄をはじめ、地形エディットやメインである都市建設モードなどの基本的操作から、用意された5つのシナリオの攻略法などを徹底解説。これ一冊で立派な市長さんになれる！

RPG幻想事典シリーズ
好評発売中!

逆引きモンスターガイド～東洋編
ヘッドルーム編著………定価1,800円

逆引きモンスターガイド～西洋編
ヘッドルーム編著………定価1,800円

戦士たちの時代
司史生/坂東いるか 共著　定価1,800円

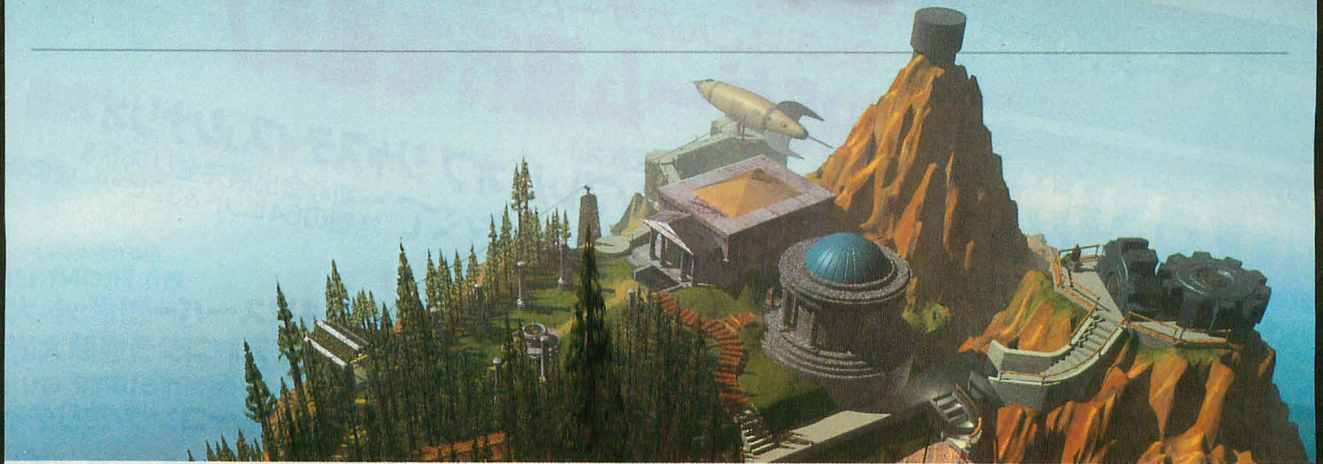
チャンバラ英雄伝
柳川房彦/高井夏生/横山雄一 共著
定価1,800円

RPG幻想事典・日本編
飯島健男 著　……………定価1,860円

RPG幻想事典
早川浩 著　………………定価1,550円

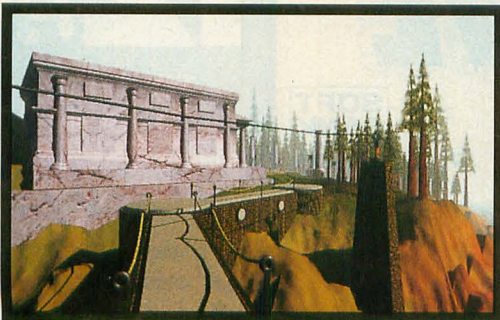
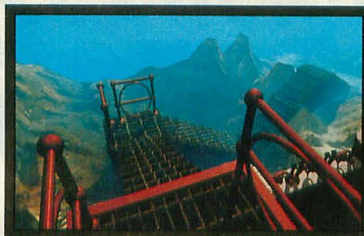
●定価は税込みです　●お近くの書店でお求めください

ソフトバンク株式会社／出版事業部　SOFT BANK
販売局 TEL.03-5642-8101

MYST™

バーチャル感覚の次世代
アドベンチャーゲーム登場!

図書館、プラネタリウム、ロケット、時計台、巨大歯車
舟の模型が沈んだ池が点在するMYST島。

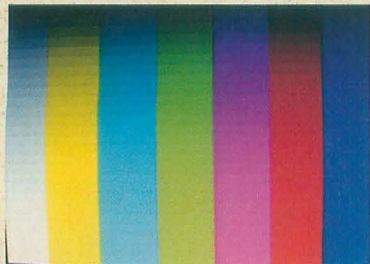
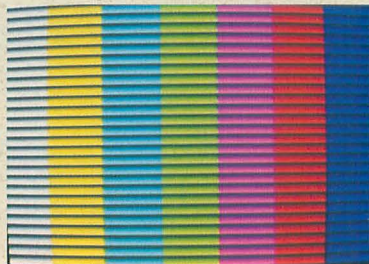
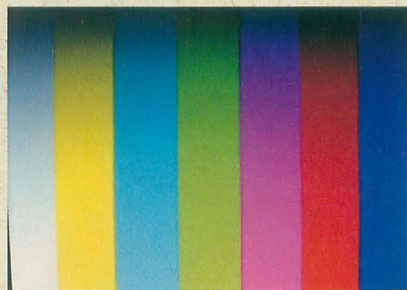
隠された謎を求めて、プレイヤーは、一冊の本を手がかりに島に
点在する数々の謎と仕掛けを、パズルを解くように解明していく。

1994年12月発売予定
SONY PlayStation版
発売元ソフトバンク
予価8,800円

深い霧の中で待ち受ける展開、予想外の出来事、
仮想世界MYST島と4つのミステリアスな
幻想の世界を冒険する
新感覚のインタラクティブムービー型
アドベンチャーゲーム。

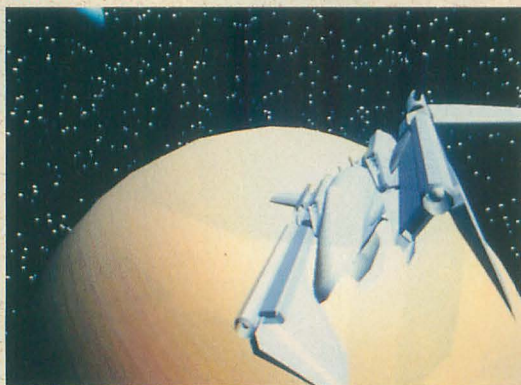
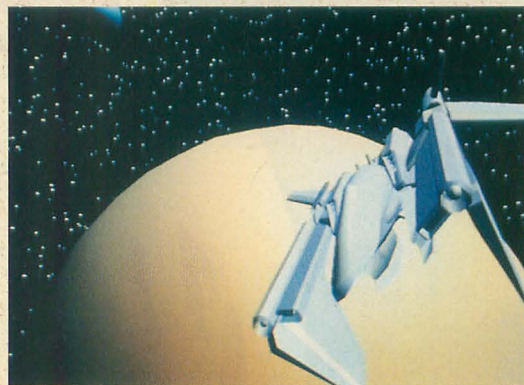
11月29日
ソニー次世代機PlayStation専門誌
THE PLAYSTATION 創刊
話題の新作情報満載!!

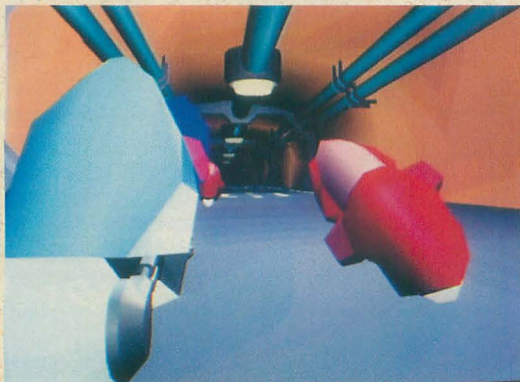
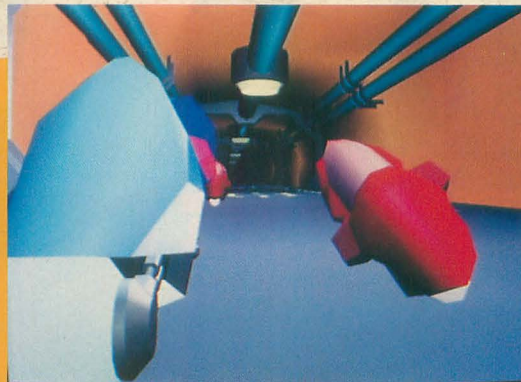
SOFT BANK ソフトバンク株式会社
東京都中央区日本橋浜町3-42-3 TEL.03-5642-8145

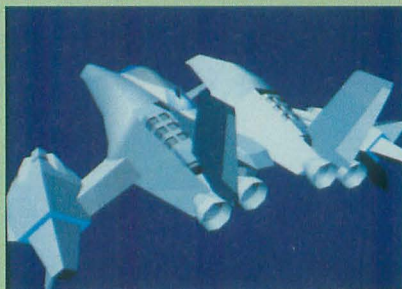
Oh!X Graphic Gallery DōGA CGアニメーション講座

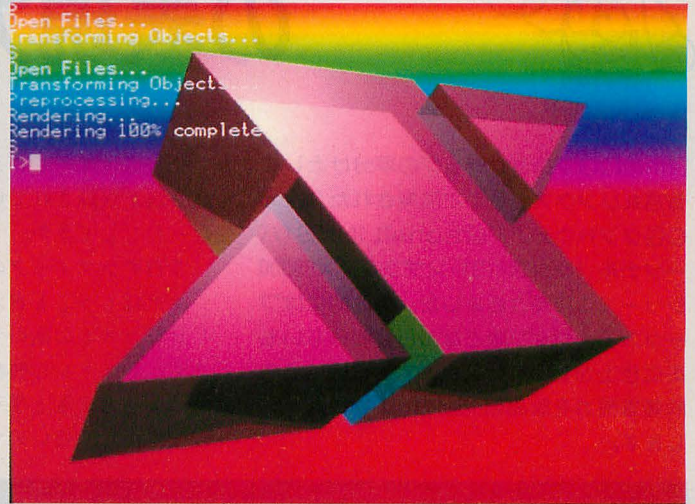
かねてより募集されている「アマチュアCGA学会論文」。その第2弾が今月号のコラムで紹介されていますので，ここでは，その参考となる写真を掲載しましょう。

1台目のX680x0で1600万色の上位ビット部分を表示

2台目のX680x0で1600万色の下位ビット部分を表示

2台のX680x0の画面表示を同期させ，スーパーインポーズ機能を使って画面出力信号を合成

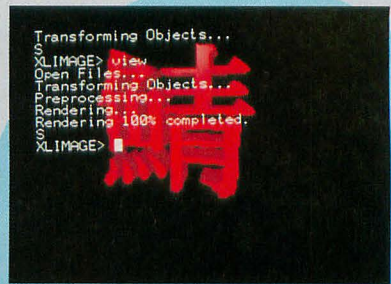
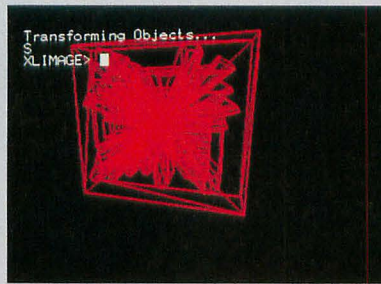
今回発表されたのは，高津正道氏による「複数のX680x0によるフルカラーアニメーションに関する研究」です。ここで提案されているシステムは，複数のX680x0の画面表示を同期させることにより色数を増やし，フルカラー表示を実現します。

表示サンプル1

3万2千色

1600万色

表示サンプル2

3万2千色

1600万色

ブースターをつけたX01号

XL/Imageお試し版＋α

正真正銘今年最後の付録ディスク。体験版つきとはいえ，残りはいつもと同じペースです。
それでは収録されたプログラムの数々をご覧ください。

XL/Image

期待の高速高画質レンダラ，XL/Imageのお試し版です。ポリゴンファイルのみという制限はありますが，それでもまだまだ強力。製品版になると表現力は格段に上がり，右のような複雑な画像も生成することができるようになります。

Modeling

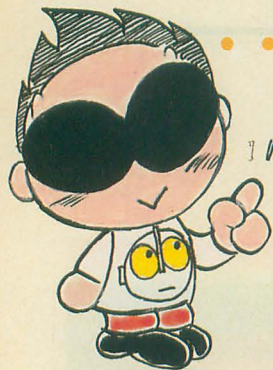
XL/Imageとお試し版で使えるモデリングツールです。回転体はマウスでチョコチョコ，ロゴ作成ツールは書体倶楽部のフォントを立体化するためのものです。

Pinball

柴田氏がコツコツ作り上げてきたものに音楽と効果音をつけて一般向けにしました。さあ，ハイスコアがあなたを呼んでいます。

SX-Pikopiko

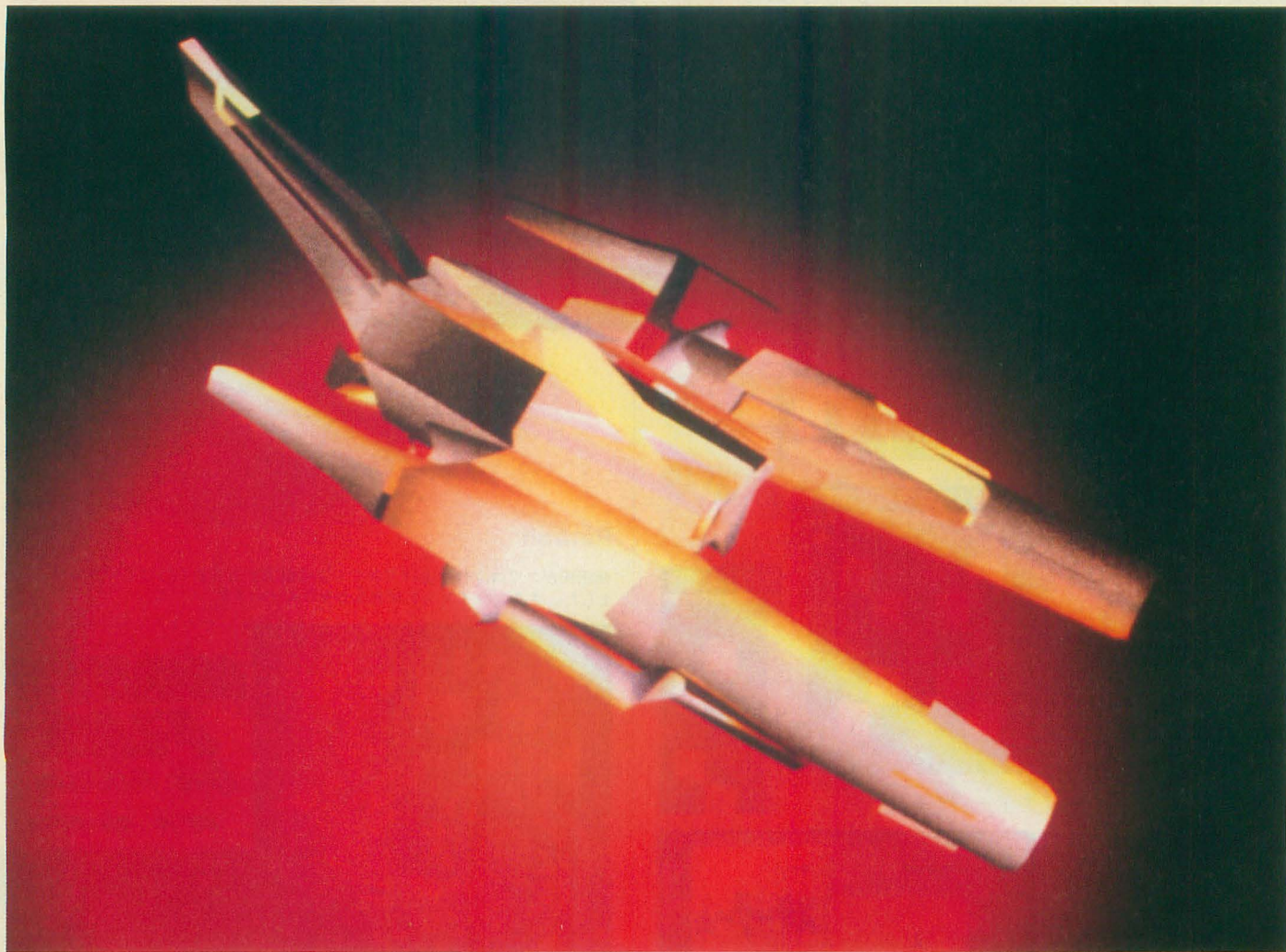
独自のピコピコエンジンを搭載し，これまでになく簡単にゲームが作成できるようになりました。懐かしの8ビット時代のVRAM構成をいまに再現したと思えばよいのでしょうか。とりあえずサンプルを見てください。

江口響子のCGれぽ～と

XL/Imageレンダラーの表現力を見る

日本のCGプロダクションの草分けとして有名なリンクス。そこで開発された3次元CGソフトウェアPersonalLINKS（パーソナルリンクス）のレンダラが，X68000シリーズに移植されました。機能的にはワークステーション版となんら変わらないものに仕上がるということです。今回はその映像表現力を製品発売前のβ版でレポートします。

今月の付録ディスクにはXL/Imageのお試し版が収録されていますが，もうご覧になったでしょうか。

3Dグラフィックの分野でもレイトレーシングはもっともクオリティの高い，しかし計算コストの高い手法として知られています。レイトレに興味を持っていても，計算時間を考えると手を出せないでいたという人も少なくないのではないでしょうか。

XL/Imageのレンダリングは従来のレイトレーサに比べて非常に高速になっています。今回掲載しているサンプル画像はすべて10MHzのマシンで計算したものですが，いちばん込み入ったものでも4時間弱で最終出力が得られました。製品版には68030＋68882専用のオブジェクトも同梱されていますので，X68030ユーザーなら何倍も高速に演算できます。PFLOAT40というドライ

この画像に使用したマシンはX68000SUPERでメモリ4Mバイト，コプロなし，外付けHD130Mバイト。画像サイズ512×512ピクセルでディスプレイに出力しています。

宇宙船のポリゴンデータはDōGAのCGA入門キットに付属のFDC.SUFをコンバータでXL/Imageに読み込み可能なPPDファイルに変換したものです。シェーディングモデルはフォンシェーディングを使い，表面は金属質感に加えてソリッドテクスチャ（後述）のノイズでバンプマッピングをしてあります。

背景には，発光型のサーフェスであるファイヤを球体に施したものを配置しています。平行光源は3，アンチエイリアシングのレベルは2です。レンダリング時間は3時間52分37秒でした。

バを使えば040turboでさらに高速に演算させることもできるようです（X68030比約2.15倍）。

そして，速度以上に重要なのが多彩な機能と出力クオリティです。

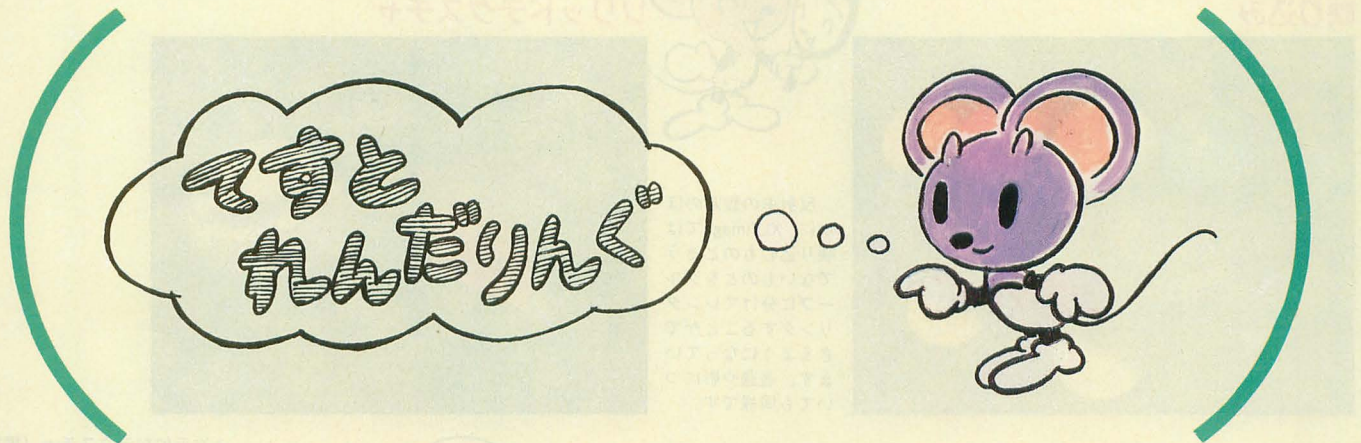
無制限に重ねあわせることのできるマッピングも強力ですし，発光体，空気遠近法，半影，デフォーカス，モーションブラーといった表現まで可能になっています。

ボリュームテクスチャを使えばパラメータをいくつか指定するだけでマッピング素材を計算で生成してくれますので，マッピング素材の大きさによらない高精度なマッピングが可能です。画像を張り込むのとは違いますからつなぎ目を気にすることもありません。

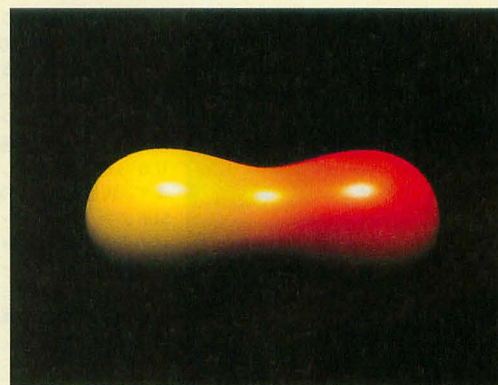
メタボールをはじめ，扱えるオブジェクトの種類が多く，これまでのレイトレーシングソフトと比較しても，シェーディングポリゴンが扱えるということで柔軟性が格段に向上していることがわかると思います。

速度と画像のクオリティでプロのCG作家の多くがPERSONAL LINKSを愛用していることもうなずけます。そしてXL/Imageは，速度はともかく，レンダラ部分の機能はワークステーション上のものに遜色ないのです（ちなみにワークステーション版は100万円以上，PC-9801版のPERSONAL LINKSは26万円もします）。

XL/Imageにはモデリングツールは付属していませんが，お馴染みのDōGA CGAシステムの形状データやZ'sTRIPHONYのデータをコンバートして使用できますのでご覧のように従来のデータを高品質にレンダリングすることができます。

では，製品版で実現されている基本的な形状や描画指定について実例を示しながら見てみたいと思います。

CSG

Constructive Solid Geometryの略。2次曲面や立方体を「和」「差」「積」の論理演算（集合演算）で組み合わせて形状を定義します。

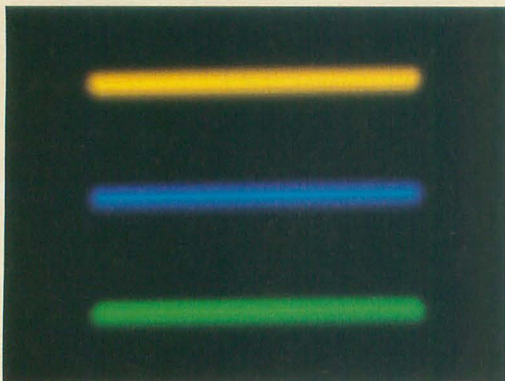
メタボール

濃度を持つ球状あるいは楕円体状の広がりのことをメタボールといいます。メタボール同士を近づけたり遠ざけたりする，濃度の広がりを変える，中心の濃度を変えることによって形状を定義します。また，色をそれぞれ別に指定すると融合部分がグラデーションになります。

負のメタボール

負の濃度を持つメタボールのこと。ほかのメタボールをへこませる働きがあります。

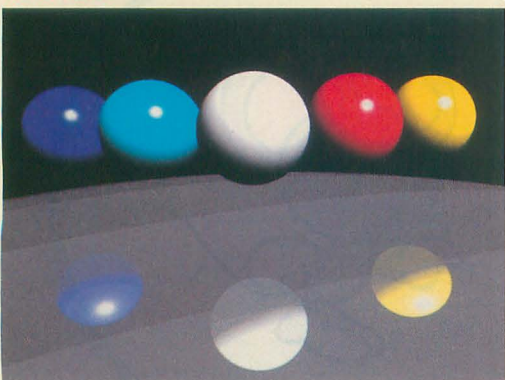
リング

厚みのない円板の形状。外径と内径のほかに，開始角度と終了角度を指定することで，閉じたリングだけでなく，開いたリングも指定できるようになっています。

ビーム

両端に球のついた円筒型のビーム形状を定義し，質感をファイヤーボールと同じ発光物体にすることで得られます。ぼんやりと発光させるのか，はっきりと発光させるのか，発光率でその度合を調節します。画像は3つとも同じ発光率を設定し，色だけを変えたものです。

ファイヤボール

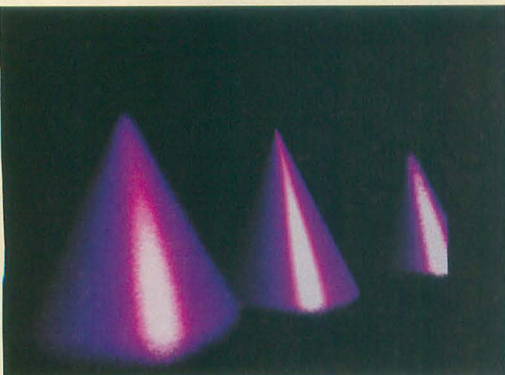
ファイヤ（発光）型のサーフェスは炎や光線など発光する物体に使います。ファイヤボールは球にファイヤ型のサーフェスを施したものです。発光物体はそれ自体が光り，ライトの影響は受けません。

映り込み

反射率の設定のほかに，XL/Imageでは映り込むものとそうでないものとをグループに分けてレンダリングすることができるようになっています。透過や影についても同様です。

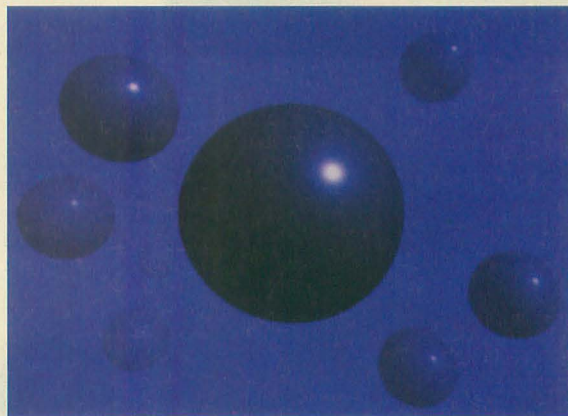
ソリッドテクスチャ

3次元的なテクスチャ（模様）。通常，テクスチャマッピングなどを球などに使うと，模様が伸び縮みした感じになります。また，マッピングされた物体を拡大すると張り付けたピクセルがガタガタに見えたりします。ソリッドテクスチャは関数で模様を作って張り付けることで，その問題を解決しています。XL/Imageには，木目，大理石のほかに，タイル，ノイズ，しわしわ，レイヤー，放射，波紋が用意されています。

スムーズシェーディング

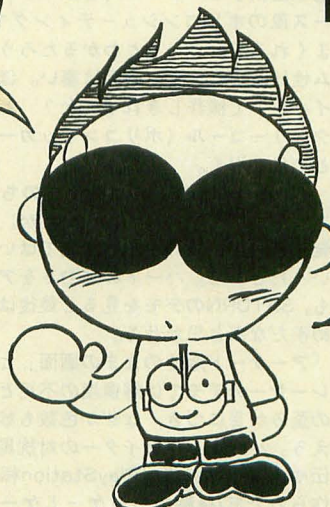
ポリゴンのがたつきをなくし，疑似的になめらかにみせる方法。フォンシェーディングとグローシェーディングの2通りから選んで指定します。画像はDoGAのSUFファイルをコンバートして読み込み，フォンシェーディングでレンダリングしたもの。質感は金属にしています。

ピンぼけ

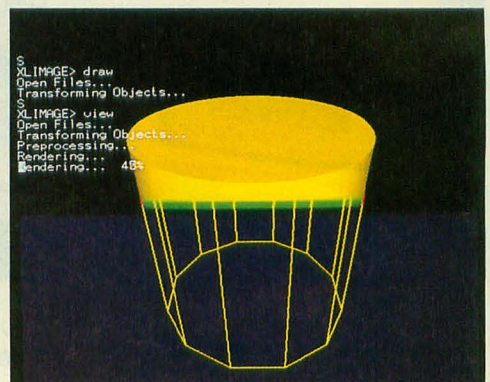
分散レイトレーシングの機能とカメラのフォーカスで得られる効果。フォーカスの合っている部分の前後はピンぼけが起こります。この画像はレイの本数を4でレンダリングしたもの。レンダリング時間は，かなりかかります。レイを増やすほど高品質になります。

フォグ

遠くのものほど霧に隠れて見えなくする手法。フォグを用いると，物体の前後関係がつきやすくなります。XL/Imageではデプスキューを指定し，物体が視点から離れるにつれて背景に溶け込む効果を表現します。

XL/Image仕様書

方式	レイトレーシング（スキャンライン，空間分割を採用）
操作方法	テキストによるコマンドファイルにより実行
形状定義	ポリゴン，２次曲面(球，楕円体，円錐，円柱)，メタボール，リング，ビーム，三角板，四角形，立方体) CSGによる定義（基本形状の論理演算によって形状を定義）
座標定義 係数定義	物体の定義は，形状定義を階層的に組み上げることで指定。階層ごとに平行移動，回転，拡大縮小，反転 ，などの座標交換や質感係数などの属性を指定可能
光源数・設定	光源種類：平行光源・点光源・円錐光源・スポット光源・環境光源 光源数　：最大32個の光源を設置可能 設定　　：個々の光源で色，強さ，対象方向，影の濃さ，および質感係数をライトに与えハイライトのできない光源にするなど
レンダリング機能	シェーディングモデル：リンクス，ブリン，フォン，ランバート 発光体の表現　　　　：発光率（中心からの減衰率で指定） リンクスモデルでは，高度な質感表現のために異方性反射，二重ハイライトをサポート，また，物体表面と視線のなす角度の変化に応じて透過係数，反射係数，ハイライト係数を変化させることも可能
マッピング	カラーマッピング，バンプマッピング，環境マッピング，ポストマッピングの４種類多重マッピングが無制限に可能。ソリッドテクスチャを用意（大理石，木目，砂目，レンガ，ノイズなどの表現）
アンチ エイリアシング	可能（処理速度との兼ね合いでレベル設定できる）
その他	分散レイトレーシング デフォーカス，ぼけた影，ぼけた反射，屈折，モーションブラー，ワイヤーフレーム表示が可能。アスペクトは，ハード制約内で任意。各種モデリングデータをサポート
解像度・出力	任意の画像解像度に設定が可能 出力フォーマット：RGBおよびα（マスク情報）の32ビット/ピクセル。
稼働環境	■ Human68k ver.2.x以上■ 4Mバイト以上のメモリ ■ 数値演算プロセッサ搭載を推奨

予価　58,000円（税別）イマジカテクノシステム　03(5449)3451

データ書式例

```
global  show-percent     on
global  background       off
global  antialias        1`

select  object  obj_root
select  camera  camera
select  background          background

create object default obj_root
          light   light1  on
          light   amb     on
          object  obj_all
close

create  object  default obj_all
        object          obj_A
close

create  object  default obj_A
        geometry        geom_A
        surface         surf_A
close

create  geometry quad   geom_A
        type    cylin
        len     1.5 1.5 1.5
close

create  surface default surf_A
        rgb     1  1    0
        diffuse 0.3 0.3 0.3
        specular 1 1 1
close

create light default    light1
        type    parallel
        str     2.5
close

create light default    amb
    type ambient
close

create camera default camera
        pos     3 3 3
        tar     0 0 0
        direction       +y
        proj-offset     0 0
        proj-size       512 512
        rend-offset     0 0
        rend-size       512 512
        step            1 1
        image           test1.lpc
close

create  volume  background       background
        rgb     0 0 1
close
```

E L E C T R O N I C S　S H O W　'94

エレクトロニクスショウ'94

次世代ゲーム機一堂に

10月4日から8日までの5日間，エレクトロニクスショウ'94が東京晴海の国際見本市会場で開催された。昨年より15％減の出展社数ということだが，見た目に規模は2まわりくらい小さく感じられた。昨年はメッセの端から端まで500mある展示場が埋まっていたわけだが，今年は晴海会場も全館は使い切っていない。

とはいうものの，今回のショウでは世間を騒がせている次世代ゲーム機が各社とも出揃ったこともあり(任天堂を除く)，会場はなかなかの盛況となった。

部門別では民生機器の活気が目立つ。特に次世代ゲーム機は特設コーナーが設けられ，SEGA SATURNやPC FX，3DO，LASER ACTIVEなどに触れることができるようになっていた。発売を間近に控えていることもあってか，SATURNの元気さが特に目立つ。

バーチャファイターは着実に仕上がりつつあり，画面の派手さではパンツァードラグーンが他を圧倒する。ゲーム内容はギャラクシーフォース風のポリゴンシューティングで360度撃ちまくれる奴，というとわかるだろうか？　ゲーム性は置いといても画面は凄い。ほかにはジョイパッドで操作しきれるのか？　と思わせるビクトリーゴール（ポリゴンサッカーゲーム）などが目を引く。

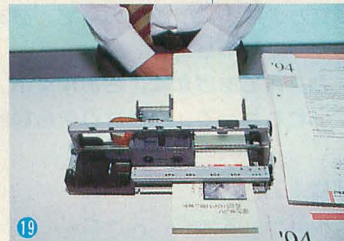
PlayStationはそのコーナーのちょうど隣のブースでデモを行っていたのだが，ソフトの開発が遅れているのか，完成度ではいまひとつといったところ。ハードの強力さをアピールしても，SATURNのデモを見ると最後はソフトが決め手だなあと思わせる。

「アーケード版そのままの画面」というリッジレーサーのデモでは解像度の不足とマッピングの歪みが目につき，なぜか色数も妙に少なく見える。バーチャファイターの対抗馬となる闘神伝やナムコの鉄拳（PlayStation相当の基板で作られた3D格闘のアーケードゲーム）を見ても，作り込みの甘さや二番煎じの感は拭えない。かえって，バーチャファイターって実はかっこいいゲームだったんだと思わせるものもある。ストIIのあとにファイターズヒストリーを見ている感じといえばわかるだろうか。

全体的にSATURNのソフトは色合いが渋くまとまっているのに対し，PlayStationのデモはどうもケバい色遣いなのが意外だった。いずれにせよ，現状ではセガの開発者のほうが魂のこもった仕事をしている。PlayStation勢ではサードパーティに優秀な人材が集まっていると思われるので発売半年後くらいが楽しみといえば楽しみ。逆にSATURNで，サードパーティにセガと同等の開発ができるかに若干疑問も残る。

それにしても，すべてのデモが横長ディスプレイで行われているのがなんとも妙な感じ。

新しい映像機器

次に映像関係を見てみよう。ビデオCDは各社からプレイヤーが出展されており，来年あたり

①元気なSEGA SUTARN ②NECのPC-FX ③ソニーのPlayStation ④サンヨーの裸眼3D表示システム ⑤三菱のヘッドマウントディスプレイ ⑥本格普及するか? ビデオCDプレイヤー ⑦ペンで押しても色が変わらない液晶板 ⑧シャープ21型TFT液晶ディスプレイ ⑨サンヨーの4倍密CD-ROMプレイヤー ⑩NECの大型プラズマディスプレイ ⑪パイオニアのMPEG2プレイヤー ⑫45万円にまで下がってきたハイビジョンテレビ ⑬エプソンのMIM液晶ディスプレイ ⑭NECバーチャル金魚鉢。だからなんだといわれても困りますが…… ⑮一部で期待のEDTV II ⑯東芝の大型LEDディスプレイ。なぜかガンダム ⑰PDは次世代メディアを統一するか? ⑱東芝のフラッシュメモリボイスレコーダ ⑲360dpi36色のカラープリンタユニット

から本格的に始動する模様。だいたい倍速CD-ROMでMPEG1のデータを再生するものと思えば間違いないが，情報量を考えればかなりの画質だということができるだろう。計算すれば1画面あたり5Kバイト程度のデータでしかないのだ。現状のCD-ROM規格とMPEG1ベースでは限界といえるだろう。

一方，サンヨーから出展されていた立体視プレイヤーは倍速倍密倍トラック仕様のCD-ROMを採用しており（データ転送速度は4倍），新たな可能性を見せている。

しかし，本命はパイオニアから発表された光ディスクシステム。CDサイズのメディアで3.44Gバイトの容量を持ち，135分間の映像をMPEG2で再生する。データ転送レートは3.4Mビット/秒と，MPEG2としてはさほどでもないが，MPEG1（1.5Mビット/秒）に比べれば格段の情報量だ。エンコードのアルゴリズムも高度になっていることだし，現行のレーザーディスク以上の画質を期待したいところだ。

このシステムの大容量化の決め手は青色レーザーの採用。

「CDサイズのメディアにLD以上の画質で2時間の映像」という要件が早くも満たされつつある。かさばって重く，イマイチの画質で保存性の悪かったレーザーディスクがデジタルメディアに代わる日も遠くないのかもしれない。

表示機器ではサンヨーの眼鏡不要の3D表示ディスプレイが面白い。

これはレンチキュラーレンズを使用したもので，キャラメルのおまけにある表面がでこぼこした立体に見えるシールをディスプレイに応用したようなものと思えばいいだろう（もうちょい複雑だが）。ちょっと見るとただのブレた画像なのだが，裸眼立体視の要領で焦点をあわせていくと，見事に立体の動画像が見える。

3D関係では三菱電機のヘッドマウントディスプレイによる立体映像などにも人気が集まっていた。ちょっと，来春発表の任天堂の立体映像マシンが気になるところ。

液晶のシャープはついに21型TFTディスプレイを展示。値段はちょっと見当もつかない。意外なところでは松下も液晶テレビデオなどを発表。「フラットパネルはやめたんですか?」と聞くと，やめたわけではないが，今回は見送ったとのこと。

そのほか，45万円というところまで下りてきたハイビジョンディスプレイや，EDTV II関連製品，昨年は目立ったデジタルオーディオ関連品も当然出展されているが，話題性はいまいち欠ける感じ。デバイス関係では各社4倍速CD-ROM用のコントローラなどを出展していたこともあり，1，2年後くらいには4倍速ドライブが標準となるのであろう。

今年のテーマはマルチメディアだが，どこを見てもまだはっきりしたビジョンに欠けている。次世代ゲーム機やネットワークへの関心が高まっていることを考えると，来年あたりになればマルチメディアの形も固まってくるのかもしれない。

［第43回］
願い事

江口 響子

響子inCGわ〜るど

ペンキの剝がれかけた１枚の扉をクリック。

すると，視界はズームインして，部屋に入る。窓のない煉瓦造りの空間のなかに，大理石のテーブルだけが置いてある。そこになにかが載っている。が，なんだかはわからない。

大理石のテーブルをクリック。

またまた視界はズームイン。今度は，テーブルのすぐ斜め上まできた。載っているのは，真鍮でできたランプと紙である。文字のようなものが書いてある。が，さっぱり読めない。

紙をクリック。

なにも起こらない。

ランプをクリック。

白い煙が立ちのぼる。ランプの精だ。ああ，これは魔法のランプだったのか。

「願い事をかなえてあげます，さあ早く！」

早くっていわれたって，いきなりそんな無理ですよ〜。

「秒読み開始！」

ええっそんな〜。え〜と，え〜と。

「３，２，１，０」

暗転。

赤い文字がフラッシュする…GAME OVER…GAME OVER…GAME OVER…

願い事をする，願をかける，というシチュエーションは，日常生活の中にそれとなく溶け込んでいます。

正月の初詣。お賽銭を投げて，今年の願をかける。

誕生日。ケーキのろうそくを一息で吹き消すと，願い事がかなう。

七夕。笹の葉につける短冊に願い事を書く。

クリスマス。サンタにプレゼントのお願いをする。

願をかけるには，ある時間が必要です。どの場合でもいきなりすぐ答えよ！　といわれてしまうと，とっさに答えるのは難しい。自分を振り返ってみると，歳をとるに従って，いまほしいものが本当に必要なものなのかどうか，ゆっくり考えるようになってきています。たとえ，それがどんなに小さなものであっても。

願い事がキーワードになっている映画が，トム・ハンクス主演の「BIG」。少年が，移動遊園地の片隅に置いてあった願い事をかなえるマシンに，『大人になりたい』と答えるところから，話が展開します。彼は，ちょっと歳上の女性に恋をしてしまい，その彼女と対等に付き合いたいと考えていたのです。でも……。

全部をいってしまうと面白くなくなってしまうので，あとは，見てのお楽しみ。

よし，再ゲーム。どこでランプの精が出てくるか，もうわかったから……。

古びた扉をクリック。煉瓦の部屋の中。テーブルをクリック。ズームイン。ランプをクリック。出た出た。

「願い事をかなえてあげます，さあ早く！」

息を吸い込む。

「秒読み開始！」

そして，答えた。

「X68000を囲むこの環境がいつまでも続きますように……」

画面が切り替わった，その先へと。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを 4 × 5 ポジで出力
作成手順
X68000でキャラクター画像（RGBファイル）を作成（使用ソフトはC-TRACE）
Macintoshでテーブル，ランプ，室内の画像（TIFFファイル）を作成（使用ソフトは，STRATA STUDIO Pro）
XIN/XOUT（電机本舗）でキャラクターのRGBファイルをX68000からMacintoshに転送したのち，Photoshop2.5JでTIFFファイルとコラージュ

SOFTWARE INFORMATION

寒くなってきました。人間はともかく，パソコンにとっては動作も安定してほどよい季節。そろそろ封印を計画している人もいるようですが，新作の誘惑の魔の手も忍び寄ってきています。さあ，どうする？

VIEW POINT

「VIEW POINT」は，レイトレーシングさながらの画像クオリティのキャラクターたちが，縦横無尽に動きまわる異次元シューティングゲームである。従来の，手描きのキャラクターがプログラマによって設定された軌道を動きまわるモノとはどこか違う。無機質なメカ生物が見せる有機的なムーブメント。なにか独特の世界観が匂うのだ。

操作仕様はショットボタンとボンバーボタンの2ボタンと8方向レバー。ショットは溜め撃ちができる波動砲方式。ボンバーは最大3つまでのストック，パワーアップはバリアとオプションのみでウェポンのパワーアップはなし。オリジナルのNEO・GEO版は，1992年に作られたゲームにしては意外にプレイヤーサイドにシビアで，実際，難易度は高かった。

ゲームの移植完成度も気になるが，音ネタを駆使したヒップホップ系/テクノハウス系のあのBGMがどう再現されるのか，はたまた再現できるのかも気になるところだ。（善）

X68000用　5″2HD版　7,800円(税別)
ネクサス インターラクト　☎03(5474)3581

「ハイパーピクセルワークス」シリーズラインナップ

1994年2月号で紹介したCGツール「ハイパーピクセルワークス」が11月1日よりバージョンアップして，Ver.2.30となった。各種の拡張ツールもTAKERUにて発売されているので，ここにまとめて紹介しよう。

＊　＊　＊

ハイパーピクセルワークス Ver.2.30　14,000円

ハイパーピクセルワークス　ライトバージョン
機能制限のある簡易版。エクステンションはすべて使用できる。　6,800円

ハイパーピクセルワークス　体験版
基本描画機能の体験版。画像の取り込み，保存は不可。エクステンションは使用できない。　200円

HPWエクステンション開発ツールキット
エクステンションプログラムの開発資料，ツール，サンプルソースコードなど。　1,200円

HPWエクステンション
- #1　マルチフォントシステム
  各種のビットマップフォント。フォントエディタもあり　1,500円
- #2　カラーコントロールセット
  画像の色を調整するツール集　1,500円
- #3　DeskJet300プリンタドライバ　1,200円
- #4　ESC/P360プリンタドライバ　1,200円
- #5　ブラシツールセット　1,500円
- #6　SHARP CZプリンタドライバ　1,200円
- #7　レインボーセット
  虹色ブラシツールと虹色グラデーションツールのセット　1,200円
- #8　BMPファイルツール　700円
- #9　ジャギー除去ツール　700円
- #10　シャドーツール　700円

X68000用　3.5/5″2HD版　価格はすべて税込
TAKERU　☎052(824)2493

レッスルエンジェルスSPECIAL

シリーズ最終作は18禁。ゲーム，グラフィックともにパワーアップが期待される。

前3作と同じくPC-98版からの移植だが，今回のX38000版はそのままのものではなく，いろいろと手が加えられている。エキシビジョンモードのグラフィックの描き直し，256色モードの搭載，サウンド変更などが行われ，さらにX68000オリジナルオープニングも追加される。

パッケージは，エクシング・エンタテインメントのブランド名で発売される。

X68000用　3.5/5″2HD版　8,800円(税別)
エクシング・エンタテインメント
☎03(5443)4967

カプコン・パソコン・スーパー・キャラクター・データ集

カプコンのイラストデータ集が再び登場した。

X68000用にはPIC形式のインストールプログラムがあるほか，多機種で使用できるように，CPI，DRA，BMP，TIFと5種類のデータ形式に対応している。

各種の案内状や年賀状向けのイラストや，カプコンのゲームキャラクターデータなど，モノクロ約400点，カラー約350点のデータを収録。

X68000ほか用　3.5″2DD版　5,800円(税別)
カプコン

パックランド

「パックランド」のオリジナルは10年前に登場したアーケード版だが，ボタン3つにレバーなしという，ちょっと変わった操作方法であった。今回のX68000版はその完全移植だが，操作の点でもオリジナルと同等に，とのことで専用3ボタンジョイパッドが同梱される。

難易度などを変更できるコンフィグレーションモードもあり。発売は12月2日。

X68000用　5″2HD版　8,200円(税別)
電波新聞社　☎03(3445)6111

発売中のソフト

★カプコン・パソコン・スーパー・キャラクター・データ集　カプコン　10/28
X68000ほか用　3.5″2DD版　5,800円(税別)

★クイーン・オブ・デュエリスト外伝α＋LIGHT　TAKERU　11/1
X68000用　3.5/5″2HD版　5,800円(税込)

★ハイパーピクセルワークス Ver.2.30　TAKERU　11/1
X68000用　3.5/5″2HD版　14,000円(税込)

新作情報

★EXEディスク2　TAKERU　11/20
X68000用　3.5/5″2HD版　200円(税込)

★上海 万里の長城　EAV　11/26
X68000用　5″2HD版　8,800円(税別)

★パックランド　電波新聞社　12/2
X68000用　5″2HD版　8,200円(税別)

★レッスルエンジェルスSPECIAL　エクシング・エンタテインメント　12/15
X68000用　3.5/5″2HD版　8,800円(税別)

★魔法大作戦　EAV　12/16
X68000用　5″2HD版　9,800円(税別)

★VIEW POINT　ネクサスインターラクト　1/8
X68000用　5″2HD版　7,800円(税別)

★X CASE　Béシステム
X68000用　5″2HD版　19,800円(税込)

★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定

★Traüm　象スタジオ
X68000用　5″2HD版　価格未定

★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5″2HD版　価格未定

★達人　KANEKO
X68000用　5″2HD版　価格未定

★エアバスター　KANEKO
X68000用　5″2HD版　価格未定

★サバッシュII　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定

★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)

★スタークルーザーII　アルシスソフトウェア
X68000用　5″2HD版　価格未定

★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定

★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)

★ディグダグ/ディグダグ2　電波新聞社
X68000用　5″2HD版　価格未定

★フォント＆ロゴ デザインツール 書家万流SX-68K　シャープ
X68000用　3.5/5″2HD版　価格未定

そこに吹く疾風のように

期待の新作「魔法大作戦」の発売日が目前になりました。まだ完全に仕上がっていないため，移植度などの細かい点に言及できませんが，サンプル版の画面を見ながら「魔法大作戦」について予習しておきましょう。

Yaegaki Nachi
八重垣 那智

唐突だがヒットするシューティングゲームというものを考えてみよう。画面はなにがなんでも縦スクロール。2人同時プレイで，プレイヤーはキャラクター選択が可能。選択できるキャラクターの中には女の子が最低でも2名欲しい(長髪とショートヘアの組み合わせ以外は絶対に不許可)。面をクリアしたら顔のアップのCGが出て，思わずハリセンでツッコミたくなるようなボケセリフを毎回かますことも，かなりおいしいポイントだ。

いうまでもなく操作は8方向レバーに2つボタンでショット&ボムの組み合わせといったベーシックスタイル。隠し味に凝るなら，溜め撃ちくらいは入れても大丈夫。パワーアップはマルチウェポン切り替え式で，全面中どこでもまったく役に立たない，心底使えない武器がないとダメ。逆にパワーアップの段階数は少なくても気にならないもののようだ。

敵の設定は細かいほうがプレイヤーへのアピール度もバッチリ。ボスだけでなく，中ボスにも細かい設定は必須。ライバル的な性格をもたせ，イメージイラストなんか用意すれば，悪役好みのファンがついてくれる。なかでも忍者のような神秘的なキャラクターは，いろいろ便利(?)なので最近では不可欠の要素らしい。

X68000用　5″2HD版9,800円(税別)
EAV　☎03(5410)3111

そしてなによりも大切で重要なこととして，タイトルには必ず漢字が入っていなくてはいけない。これを忘れたら，全部オジャンになると思ってもいいだろう。と，このようにここまで書いてきたことをやれば，そのゲームは大ヒット間違いなしのハズなので，もっと縦シューが出ないかなぁ，などとモノ思いに耽る今日この頃である。

破壊の傾向と対策

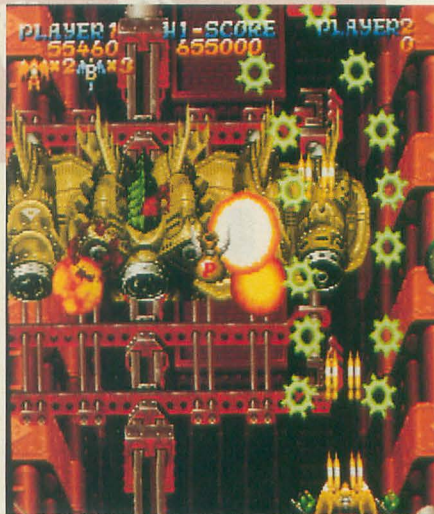
前置きは冗談としても，最近見ることのできるシューティングゲームは多かれ少なかれこういった傾向に基づいて作られている。今回紹介する「魔法大作戦」は，こういった売れセンをかなり意識したというより，そういった流れを決定づけたゲームであるという印象をもたれることが多い。オリジナルが，そのライジングという耳慣れない社名と同時にデビューしたのが1993年の初夏である。ほぼ同時期に「戦国エース」がデビューして，これがまた前置きに書いた特徴に忠実だったものだから，この2つのゲームは多くの人に，流行の縦スクロールシューティングというイメージを擦り込ませるのに，十分な威力を発揮したといっても間違いないだろう。

ハードの性能や容量で遥かに勝っているハズの，1994年前半に出た一連の昔ながらのメカフェチ硬派シューティング(あえて名前は出さない)が，いまいちパッとしなかったのはこのあたりに理由があるのではないかと思い込みそうになるくらいだ。

そこそこに派手な攻撃やグラフィック，クセのあるプレイキャラクターたち。このゲームの特徴は最初に並べたとおりなのだが，あえて書くとライバル役として登場する赤いエビのような中ボス「バシネット」の存在感が一番の目玉だ。独特のBGMと一緒に目と指に刻み込まれる感覚は，ほかにはあまり例を見ない。「戦国エース」よりも緻密な印象を受ける理由は，ひとえにこういった作りの丁寧さによるものが大きいのだと考えられる。

そういった演出や雰囲気を楽しむといった要素の強い最近のシューティングの代表の1つである「魔法大作戦」が，このほどX68000に移植される運びとなったのは，格闘とレトロゲームに話題が偏っていたX68000にとっていいことではないかと思う。筋肉とムチムチの格闘ゲーム漬けの脳ミソと指先でお悩みの人のために，今回はサンプル版を見ながら「魔法大作戦」の世界を予習していくことにしよう。

戦場の掟と約束

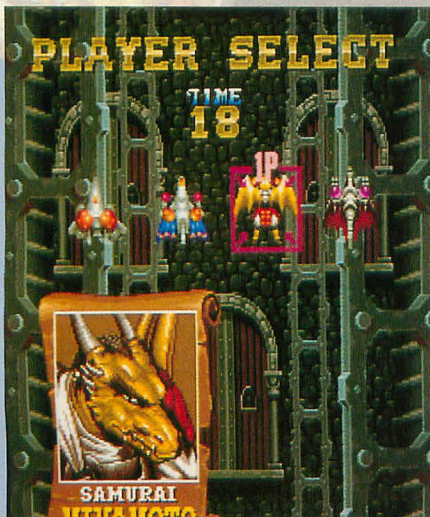
このゲームの操作から説明すると，移動は8方向で攻撃は2ボタン。ボタンは説明無用の通常ショットと緊急回避型ボンバーで，シンプルそのものである。プレイヤーが選べるキャラクターは，やや少なめの4種類。キャラクターは移動速度と武器がすべて異なり，好みで選べるように一長一短があるため万能キャラに偏ったりすることがない設定には，手慣れた印象を受ける。また，武器が異なるといっても各アイテムに対応する攻撃のグラフィックなどが異な

グラフィックは完璧だ

るものの，パワーアップシステムそのものは共通だから，混乱することはまずない。

ただパワーアップは，メインショットとサブショット(魔法)が独立していてそれぞれを運んでくるキャラクターも異なっているので，ちょっとややこしい部分がある。

細々と書くと，このゲームのパワーアップアイテムは，基本的に3種類あり，メインショットをパワーアップする金色のコイン，サブショットをパワーアップする本，そしてボム補給アイテムと区分できる。

コインは羽の生えた袋を攻撃することで4枚のコインがバラまかれ，それを回収していくと徐々にメインショットの攻撃力がアップしていくようになっている(勘のいい読者はなにかに似ていることに気がつくだろうけど，いわないのがオトナのオヤクソクというものらしい)。コインは画面左右に散らばって，比較的速いスピードで落下していくために，スピードの遅いキャラクターは，なかなかメインショットがパワーアップしないようになっている。こういった部分でキャラクターの差が出るようになっているところには，ちょっと注目しておいたほうがいい。

サブショットアイテムの本は，特定の場所で妖精が箱に入れてもってくるようになっている。その内容によって3種類の色とアルファベットで区別され，同じ色を続けて取ればレベルアップし，違う色を取ったときはサブショットのチェンジとなる。前方攻撃のF(緑)，拡散攻撃のW(青)，追尾攻撃のH(赤)と，どれも便利そうだが実はキャラクターによってこれらの具体的な攻撃方法や得手不得手までが異なっている。そのためどのキャラクターでも通常はWでボスにはFというような単純な仕組みにはなっていない。

ガイコツの伸びてくる腕がいやらしい

派手な炎が美しい

最後はボムアイテムで，特定の敵を破壊したりすると出現する。ボムは全キャラ共通で，性格的には緊急回避と集中攻撃の両方の要素を兼ね備えているが，効果が出るまで，やや間があるので先読みして使う必要がある。しかもボム炸裂中はサブショットが使用できなくなるので，ボムを使って近づいて連射する戦法が，思ったほど効果がないのが残念だ。

こうしてプレイヤーのシステムをまとめてみると，イメージとは裏腹に練り込まれた本格的なシューティングの匂いを感じるだろう。これらの武器を駆使し，残機のある限り敵を叩き，そして破壊する。悪のゴブリガン帝国の野望を打ち砕くまで戦いは続くのである。

己を知り，勝利せよ

ゲームシステムが説明できたところで簡単にプレイキャラクターたちも紹介しておこう。こういったゲームでのキャラ選択のポイントは，自分にとっての扱いやすさだが，つまらない部分に惚れて苦手なキャラを極めてみると意外と楽しい発見が多いので，ちょっとオススメしておこう。

●ガイン(戦士)

なんかいつも怒っているような気がするサルつきのおにいちゃん。これといって極端な特徴がないのを，平均的で扱いやすいと見るか，平凡で見どころがないと感じるかが分かれ目か。基本はFの剣状ショット。

●チッタ(魔法使い)

女の子。都会にあこがれているところを見ると田舎のダサイおねーちゃんにしか見えない。速度は比較的遅いが，Hの火炎噴射ホーミングのようなハデな攻撃は見ていても頼もしい。ただ敵弾が見づらいかも。

●ミヤモト(侍竜)

見かけからしてドラゴン，しかも侍。自らの翼で飛び，師匠の仇であるツムジ丸という忍者を求めて戦場を征く。スピードがあるのでメインショットのパワーアップは楽。集中攻撃力はあまりないが，Wが炸裂し爆風に敵を巻き込むので便利。

●ボーンナム(呪術師)

珍しい骨を集めるためになんでもする怪しいジジイ。速度は出ないが，Fを取ったときの攻撃は貫通弾にもなり最強だ。さらにメインショットのパワーアップさえ耐えれば天国だ(?)。意外に使い勝手はよいはずなのだが。

結局はどれを選ぶかではなく，思う存分遊べるのだから，全部のキャラクターをひととおりプレイしてみるのがいいだろう。目指すは全キャラでクリアといったところだろうか。

希望はもっと遠くまで

今回は「魔法大作戦」の紹介ということで，あくまでも移植の内容には触れないで誌面が尽きてしまったが，この原稿を書いている時点のサンプル版では，ひととおりのデータが入っておりプレイすることも可能だった。見たところ，グラフィックはオリジナルからほぼそのままコンバートされているようだ。

現状で個人的に感じた様子では移植の感覚は悪いものではない。おいしい演出や，外道な仕掛けや攻撃も健在。今回の移植で，魔法大作戦を堪能できるのではないかと思われる。少なくとも，完成版に大いに期待できるレベルだ。発売まであとわずかなので刮目して待つとしよう。

プレイキャラクターはどれにする？

THE SOFTOUCH

●スーパーストリートファイターII

トーナメントモードに燃えた夜，の巻

Nishikawa Zenji
西川 善司

格闘ゲームのなかでも，室温が急上昇するのが人間同士の闘いですが，X68000が1台あれば100円玉がなくても8人で遊べちゃうのがスパIIです。Oh!X編集部にも「優勝」の2文字を心に秘めた人々が集まりました。

みんな「スーパーストリートファイターII」はもう買ったかな。まだの人はいますぐソフトショップへGo！6ボタンスティックがない人はそれも一緒にな。

スーパーストリートファイターII(以下スパII)にはトーナメントモードという一風変わったゲームモードがついているのを知っているかな。そんなに広くないゲームセンターの中に所狭しと横に並んだ4台のスパII筐体は，まさに留まるところを知らないカプコンパワーの象徴だった。

1試合は1本勝負。通常の対戦ならば，もしも1敗してもそのあと2勝すれば「勝ち」だったんだけれど，トーナメントモードで「優勝」するためには，1敗もできないということなのだ。チビシィィィー。継続してプレイができるのは優勝者のみ。敗者7名は筐体の前から去らなければならないのだ。

実はこのシステム，同じ8人で遊べる乱入台やひとりプレイ台よりもゲームセンターは儲かるんだよね。再プレイする場合，トーナメント台では優勝者以外の7人が100円を入れることになるわけだけど，同じ8人が遊べる4つの乱入台は4人の敗者が100円を入れることになる。単位時間あたりにすぐ次のプレイヤーがプレイすると仮定すれば，トーナメント台では一度に700円の儲け，乱入台では一度に400円の儲けってことになる。ウムム，さすが商売上手のカプコン。

で，このトーナメントモード，なんとX68000版のスパIIにも搭載された。8台のX68000が必要??……いや，そんなことはない，安心されたし。1台のX68000で8人のトーナメントができちゃうのだ。

8人の参加者それぞれが自分のプレイヤー番号とプレイキャラクターを選択すると，コンピュータが対戦相手をランダムに決めて「プレイヤー3は左の席へ，プレイヤー7は右の席へ」というように画面で指示をしてくれる。1位を決めるだけでなく，3位以下のすべての順位を決める試合までやってくれるので，プレイヤーはどんなに負けても3試合はプレイできることになっている。また，トーナメント参加者が8人に満たない場合，残りのプレイヤーをコンピュータが担当してくれる。ちょっと寂しいけれど，ひとりだってトーナメントモードで遊べるってことだ。

集え！ストIIバカ

んで，このトーナメントモードを使って遊んじゃおう，ってことでもの好きなOh!Xライター陣は去る10月1日，Oh!X杯争奪スパII大会を開催したのである。参加者は瀧康史，山田純二，丹明彦，朝倉祐二，高橋恒行，菊地功，中野修一，豊浦史子，須藤芳政，古村聡，江川乃誉司，そして私，西川善司の12人(順不同/敬称略)。ぐわぁ，狭いマシンルームに集まった集まった，意味不明にギラギラした目のこの連中。ほかにやることぁないのくわぁーって，こりゃあ，幹事やってる私がいう台詞じゃないか。

X68000用　5"2HD版　9,800円(税別)
カプコン

まずは第1戦

トーナメント第1回は，いわゆる得意キャラによる対戦とした。このファーストバトルの参加者は瀧，山田，丹，菊地，中野，須藤，古村，西川の8人。キャラクターはケンが大人気で，瀧，山田，丹，菊地の4人がケンを選択した。ケンは昇龍拳，波動拳をマスターできれば，意外に初心者/熟練者の力の隔たりは少ない。特に，相手方向につんのめるように出る昇龍拳はあらゆる状況で対戦相手にとって脅威となる。このあたりが人気の秘密か。

中野氏はブランカを「まぁ，私は……」と選択。氏は初代ストIIからの筋金入りのブランカ使いだ。

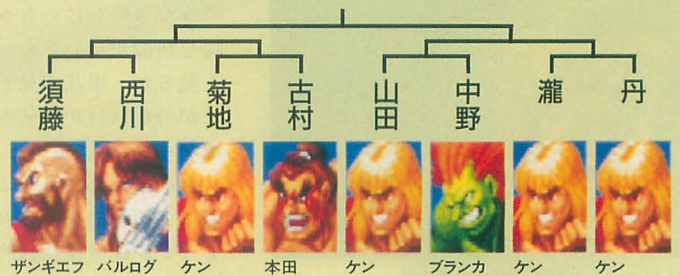
須藤氏は「えへへへ」とニヤニヤしながらザンギエフを選択。彼のザンギエフに畳み込まれたら最後，転ばされるか掴まれて宙を舞うかのどちらか，という話。

古村氏は「まぁ，私は本田さまですねぇ」。絶えず「本田さま」と呼ぶあたり相当な本田フリークらしい。

そして私，西川善司は「美しい私には鮮血の赤が相応しい」と呟きバルログを選択。

各プレイヤーが使用キャラクターをX68000に登録すると図1のようなトーナメント図が表示され，闘いが始まった。

第1試合は私と須藤氏。須藤氏は不気味な笑みをこぼしながらスティックを握り，早くもグルリグルリとスティックを回し始

図1　トーナメント表

めた。
須藤「へへへ，私の好きですアタック受けてくださぁいぃぃ，ウキキ」
と須藤ザンギ，突然のボディプレス。
西川「ぐを」
着地と同時にペシペシと小パンチが数発入ったあと「フン」のザンギのかけ声。西川バルログは掴まれて空中へ。その後も，西川バルログは跳び込めばラリアットで叩き落とされ，バルセロナアタックを出そうと壁を蹴れば頭突きで気絶させられ，まったくイイトコなし。最後は顔面マッサージ掴み技で昇天させられた。合掌。西川バルログ1回戦敗退。須藤ザンギ2回戦進出。
須藤「ウキキ」
まったくこの人の言動には謎が多い。
第2試合は古村本田対菊地ケン。
菊地「……」
古村「私の出番ですかぁ」
古村氏は古村スマイルを浮かべプレイヤー席へ，菊地氏は無言のまま席へ。プレイ開始。
菊地「……」
無言の菊地ケンは執拗な攻撃で古村本田を責めたてる。動きの重たさと飛び道具に対抗する術のない本田は苦戦を強いられる。「本田さまぁ……」古村氏の悲痛な叫びも画面のなかには届かない。菊地ケンは波動拳で牽制，昇龍拳で叩き落とす完璧な戦闘パターンを展開。一発の攻撃力の大きさがあなどれない本田も攻めのきっかけが摑めない。完璧な攻防の菊地ケンの前に古村本田は惜しくも敗退した。
古村「本田さまぁ，しっかりぃ」
菊地「……」
菊地氏は無言で拳を天空へ突き上げた。
第3試合は，山田ケン対中野ブランカ。山田ケンは守ってよし攻めてよしのスタンダードな実力を備えている。特別破天荒なプレイは行わないが，かといって地味でもない。あなどれない存在だ。中野ブランカ

愛の本田様。おしりでやーん。

昇龍拳は上昇中は無敵だ!!!

なすすべなーし！ のよわよわバルログ

は先述のとおり歴史がある。名勝負の予感。
山田「ハハ，なんかやだなぁ」
ちょっと緊張気味の山田氏。
中野「まぁ，スーパーのブランカですから……」
一同「……」
謙遜なのか余裕なのか，中野氏の玄人な発言に一同ただ納得。ゲーム開始。
中野ブランカはガンガン通常技→必殺技の連携攻撃を繰り出して山田ケンを圧倒。上段下段攻撃を織り交ぜたトリッキーな攻撃に山田ケンは翻弄される。しかし，ラウンド中盤に決まった山田ケンの1発の飛び蹴りをきっかけに，ゲームの流れは一気に山田氏へ。
中野「あっ，ダメだというのに」
後半からリズムを崩した中野氏劣勢の様子。ゴスゴスとしゃがみ防御の上から大昇龍拳で削る山田ケン。結局，山田氏の勝利。
山田「ハハ。やったぁ」
喜び方もスタンダードな山田氏だった。
中野「スーパーのブランカですから……」
1回戦最終試合は，瀧ケン対丹ケン。同キャラ対戦だ。瀧ケンはおたっきーケンともいわれ，連続技やらキャンセル技やら，何かとハイレベルな技を出したがる癖がある。そこが弱点でもあるが恐ろしいところでもある。
瀧「知ってます？　ケンはこの跳び込みパンチから……(中略：延々と連続技の解説が続く)……のキャンセルから昇龍拳に繋がるんですよ」
一同「……」
一方，丹ケンはとっさの昇龍拳が完成度70%という，ちょっと粗削りなところが気にかかる。同キャラということもあり，丹氏はどことなく不安気な面持ちで席に座る。
丹「もう始める前に勝負が決まっているような気が」
いきなり弱気な発言の丹氏。
波動拳の撃ち合いで始まったこの試合は，お互いの必殺技が火花を散らしぶつかり合うド派手な試合となった。ただ，おたっきー瀧ケンのほうが，連続技を確実に決めている分，丹ケンより優勢。
瀧「知ってました？　この波動拳を……(中略)……まで入るんですよ」
試合中に，自ら自分の繰り出した連続技の解説を入れてしまうあたり，余裕が感じられる。結局，試合は瀧氏の勝利。
瀧「いまのは大足払いからのキャンセル……(以下略)」
丹「あははは」
解説は止まらない。丹氏はなぜか負けて笑っている。

2回戦

2回戦の第1試合は須藤ザンギ対菊地ケンの対決。
菊地「……」
須藤「ウキキ，ウキ!!」
静と動という言葉では片づけられない，なんか妙な取り合わせだ。
ゲーム開始後，菊地ケンは波動拳を連発。そう，ザンギ，それもA級ザンギと戦うには間合いを離して戦うことが必須事項。詰められたら最後，グルグルドシーンの1ナイトトリップが待っている。
須藤「ウキ，きびしぃー」
須藤氏は財津一郎の物真似をしながら，飛び道具すり抜けのダブルラリアットで波動拳をかわしつつ間合いを詰めようとする。が，菊地ケン，容赦ない足払い，昇龍拳で接近を許さない。
須藤「ウキキキー」
須藤氏，自慢の長髪をかきむしりながら悔しがる。菊地氏の勝利。
菊地「……」
2回戦第二試合は，山田ケン対瀧ケン。
山田「ハハ，なんかやだなぁ」
今回も緊張気味のようだ。山田氏，セリフも前回と同じ。

スーパーのブランカですから……

瀧「また，ケンとの対戦かぁ」

みんな同キャラ対決となると緊張するようだ。キャラクター間の有利不利なしの，本当にプレイヤーの腕前が試される対決となるからか。

プレイ開始。いつもなら開始直後に波動拳を撃ち合うパターンなのに，今回はいきなり後退ジャンプして間合いを広げてからの波動拳。お互いかなり意識している様子。

と，ジャンプから突然竜巻旋風脚を出して間合いを詰める瀧ケン。波動拳の連射で山田ケンを画面端に追い詰める。

瀧「汚いですよね，こういうの。でも，ここから……（中略）……出せば抜けられるんですよ」

瀧氏，いきなり「鳥籠状態」突破法を解説。

山田「えい」

突破法が役に立ったのか山田氏反撃！そのあとのすさまじい攻防は筆舌に尽くしがたい。結局僅差で瀧ケンの勝ち。

決勝戦

決勝戦はありゃりゃ，瀧ケン対菊地ケン，またケン同士の対決。まぁ，参加者の半数がケンだったんだから仕方がないか。

瀧「ぼくは，1回戦からずっとケンとの対戦ですよぉ」

菊地「……行きますよ」

沈黙のファイター菊地氏がついに口を開いた。な，なにかが起こる!?　ゲーム開始!!

波動拳！

菊地ケンが開始直後に波動拳の速撃ち一発。瀧ケンは垂直ジャンプで一瞬のうちにこれをかわし空中から竜巻旋風脚。意表を突いた瀧ケンの先制攻撃ヒット！　と思いきや，

昇龍拳！

菊地ケン，瀧ケンのトリッキー攻撃をなんなく撃退。一瞬のうちに高度な読み合いの試合が展開され，一同賞賛のうなり声。さすが菊地氏。珍しくポーカーフェイスが崩れニヤリ。西川はポカーンフェイス。

瀧ケンは間合いを広げて遠くから波動拳で菊地ケンを牽制。そこからジャンプキック。ニヤリと菊地ケン。落ち際を，

昇龍拳!!

瀧ケンが吹っ飛ばされ……ない！　さすがはおたっきー瀧ケン，昇龍拳の射程外ギリギリに着地。菊地ケンは空しく昇龍拳ポーズのまま天空へ。拳が無意味に燃えているのが悲しい。菊地ケンの着地を瀧ケンの毒牙が待ちかまえる。

菊地「ウワ」

華麗な連続技が炸裂。

瀧「いまのはですね。アッパーからの連続技で，スーパーになってから決めにくくなったんですよ」

瀧氏の解説も炸裂。連続攻撃を食らった菊地ケン一気にピヨピヨ状態。解説を食らった一同もピヨピヨ状態。さらに瀧ケンは連続技を非情にも菊地ケンへ叩き込む。

瀧「これはさっきのと違った連続技で……（以下略）」

瀧氏容赦なくギャラリーに連続解説攻撃。

菊地ケン「うーわー」

一同「うーわー」

スパII楽しみ方いろいろ

Oh!X杯争奪スパII大会は，見事，瀧氏のおたっきーケンの優勝で幕を閉じた。

古村「げげ，これってもしかして1位，2位がケンてことじゃない」

須藤「ウキキ，3位は私のザンギです。4位は山田さんのケンです」

西川「ケン陣は頑張ったねぇ。あれぇ，もうひとりのケンは？」

丹「うるさいなぁ」

西川「おや，丹さんのケンはビリですね。頭と尻がケンというわけですな。ガハハハ」

丹「ブービーの善ちゃんにはいわれたくない」

一同「どわはは」

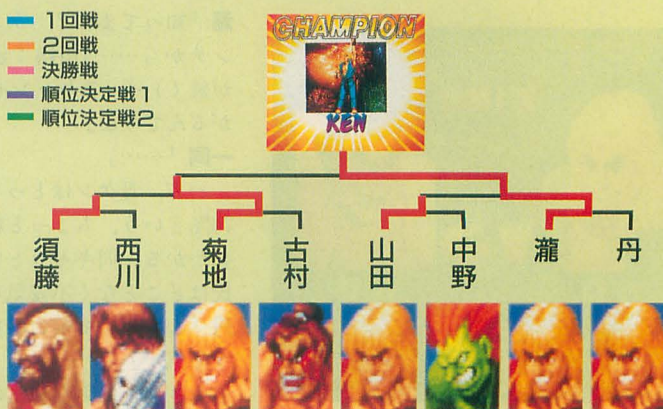
トーナメントの結果は図2に示しておこう。

この後，山田氏の作成したランダムキャラクター割り当てプログラムによるトーナメントが行われた。これは参加者の意志に関係なくキャラクターをランダムに割り当てて対戦を行うもの。普段使い慣れないキャラクターに当たれば苦戦は確実。しかし，プレイヤー間の実力の差も適度に縮まり，名勝負，珍勝負のオンパレード……になるはずだったが，皮肉にもケンが瀧氏に割り当てられ，彼が再び優勝。あーぁ。

さらにこの後，各自得意キャラを3つまで選択，負けたほうは次のキャラクターを出していき，持ちキャラを使い果たしたら負けという，柔道ルールによるスパII大会も決行。全員参加の総勢12名による大トーナメントとなった。こちらはトーナメントの管理を人間がやらなければならず，大変な時間を要した。優勝者はえへへへ，私，西川善司。正義は最後には勝つのだ。ガハハハ。

※掲載した写真はすべて後日再現したものです

図2

	瀧	菊	須	山	中	古	西	丹
第1位　瀧		○		○				○
第2位　菊地	×		○			○		
第3位　須藤		×		○			○	
第4位　山田	×		×		○			
第5位　中野				×		○	○	
第6位　古村		×			×			○
第7位　西川			×		×			○
第8位　丹	×					×	×	

特別編5

いよっ! 日本一のイケイケお嬢!

Mai Shiranui

Komura Satoshi 古村　聡

揺れるぅー，重いぃー，身体中感じてー。ぷるんぷるん。エー，最近ゲームセンターに登場したMVS/NEO･GEOの最新ゲーム「King Of Fighters'94」なんかではもう，それはバケツいっぱいのイチゴ味ゼリーのように，天井にぶら下げた夕張メロンのように，もうぷるんぷるんの舞ちゃんなんであります。シリコンはガンになるから気をつけましょうね。

「餓狼伝説2」から舞ちゃんは，扇子を持ち，服はなつかしのジュリアナ系ディスコイケイケファッション(やっぱりあれって一種のボディコンだよねねねっ)。大仏は沈んでるし，いまだにいかだを使っているし，とても日本とは思えないド田舎に住んでいるとは思えないほどファッションセンスのよさが光っていたんでありますが，餓狼伝説SPECIALではさらによい！　そう，キャラクターの色が2色のうちから選べるようになったんですけど，この舞ちゃんの新色「青」がすっげーきれいでいいですよねー。うーん，いいよなぁ。実は私，前はアンディ使いだったけど，この青舞のおかげで舞ちゃんを使うようになったんです(あ，クラウザー様も使ってるけど)。

いや，本当はそれもこれもアンディーが悪いのだ。まーったく小斬影拳は弱くなるしさー。それにあの新色の黄色！　ちょっとセンスが悪すぎる！　舞の弟子になって，ちっとセンスの勉強したほうが……。でも，舞ちゃん使いになってしまうと今度は，エンディングでアンディーと舞がくっつくとこ見るのもちょっと妬けてイヤなんですよね。うーむ，揺れるおとこごころ。どうりで，餓狼伝説2でアンディー使ってるとムキになってかかってくる舞ちゃん使いが多かったわけだ。こーの，身も心も男を悩ます日本一のラブコメ娘っ。くうーっ。

龍炎舞でOK☆

で，まぁ，そんなわけで，なんとか私もエンディング見られましたんで，攻略法を……って，実は私，1回エンディング見ただけなんですけど……。しかもEASYで。

えっと，舞ちゃんの持っている必殺技はクルリと回ってしっぽが燃える龍炎舞，ジュリ扇飛ばす花蝶扇，SPECIALではどの面でも使えるようになったムササビの舞，体重を預けて体ごと飛んでいく必殺忍蜂，それと超必殺技のわりにはコマンドの簡単な超必殺忍蜂とあるわけなんですけど，なんといってもやっぱりいちばん使える技は龍炎舞なんでありますよ，龍炎舞！　煎餅ジジィ山田スケベェがすり足で寄ってきたら，シッポでバシぃ！　空飛ぶビヤダル，チンさんもバシっ！　大のほうの龍炎舞は当たると相手のダメージもすごく大きいうえに，対空技としても使えるんでありますね。

そうそう，SPECIALでは2と違って，攻撃して敵がひるんでるすきにすばやくバシバシと連続で技を入れちゃう，いわゆる「連続技」っていうのもあるんですけど，舞ちゃんではこの連続技も龍炎舞はポイントになるんですよね。近距離でのキックは攻撃が速いんで，キックの直後に龍炎舞をすればバシ，バシバシと3発分の攻撃が入るってーわけなのです。

やっぱり「ぷるるん」だよなあ

で，これだけいろいろ使えるお得で便利なコマンドが，ジョイスティックをテンキーでいうと↓↙←とぐるっと回すだけ。溜める必要もないし，スティックを4分の1回転するだけなんだから，やー，舞ちゃんってほんっとにいいですよね(いや，ただ私が昇龍拳系のコマンドが苦手ってだけの話なんですけどねー。でもユリちゃんの覇王翔孔拳は使えるぞ。えっへん)。

本当は確か，立ち大キック→大龍炎舞ってのも入って，威力が大きいって聞いたんだけど，私がヘタなのか，ちっとも入らないんですよねー，うーむ。もしかして，X68000版はNEO･GEO版より入りにくくなってるんだろうか？　(んなことないってば)

コダワリの女の子キャラなのだ

そういえば，舞ちゃんもそうなんだけど，格闘ゲームってどれも女の子キャラの技のコマンドって，使える技がほかのキャラに比べて，わりかし楽に出せるような気がしませんか？　ナコのアンヌムツベといい，キングのベノムストライクといい……。

え，何？　舞を「ちゃん」づけしたり，キングを呼び捨てにしたりするなって？　いーのっ。舞ちゃんはやっぱり舞「ちゃん」だし，ゆりはやっぱりユリちゃんだけど，春麗はやっぱり春麗だし，キングとシャルロットは様をつけるけど，ナコルルはナコなのっ。ナコルルはときどき「ねこるる」になったりしてもうひとりたちるるがって，なんかほとんど同人誌ネタになったところで，これにて劇終。

首筋にちゅっ！　許せん，アンディー

舞ちゃん，もえもえー！

特別編6

ザコは寝ていろ Geese Howard

Taki Yasushi 瀧 康史

ギース様は無敵なり

「餓狼伝説」のエンディングで，テリーたち3人組に自分のビルの最上階から落とされて殺されてしまったギース。ところがSPECIALでは見事な復帰。一説によれば，地上に衝突する寸前，烈風拳を地面にぶち当ててその反動で生き延びたという(それが疾風拳？)が，定かではないところ。

一度は悪役として出てきても，敗北したら主人公たちの味方という，軟弱なやつらは結構多い。しかし彼は何度復活しても悪役という，世界中の悪役のお手本ともいえるべきヤツなのだ。これがギース様の魅力の根源といえよう。

当て身投げの美しさ

餓狼SPでは，全般的にキャンセルによる連続技をつなげてプレイするキャラクターが多い。ギース様も同様，キャンセル技は確かにあるのだが，必殺技がうまくつながらない場合が多く，結局，削りや牽制の範囲を出ないのだ。よく使う牽制技といえば大キックからの小烈風拳などだろう。

中段当て身投げは狙うのが難しいぞ

超裂破弾だって，上段当て身投げ

竜虎乱舞は当て身投げできない

そんななかで，ギース様のよさを最大限に発揮するのが当て身投げだ。コンピュータ相手に食らったことがあればわかるだろうが，当て身投げというのは，ひと言でいえば相手の攻撃をすべてキャンセルし，自分の投げ技につなぐという技だ。相手が無敵技を出していても，自分の投げ技につなげてしまうので，ザコにとっては屈辱で，悪役としては最高に気分のよい技なのだ。

というわけでギース様を使わせていただくならば，当て身投げを極めなくてはならない。当て身投げはコマンドを入力完了したのち，グラフィックが変わっている間に敵の攻撃を受ければ，そのまま投げにつながる。この当て身投げには2種類，小キックで出る上段(および必殺技用)の当て身投げと，大キックで出る中段(いわゆる地上の通常技)の当て身投げがある。跳び込み技，必殺技，超必殺技は上段なので，主に使うのは上段当て身投げだろう。愚かしくもギース様に攻めるザコは，跳び込みに攻撃をまったく出さずに，着地したところを投げようとするときがある。こういうザコには立ち小キックを当ててやろう。

鳳凰脚だってこのとおり

ギース様御乱心

中段当て身投げは，ふつう，地上での通常技を当て身投げする。といっても，身近でギース様相手に，小技を出すザコはいないだろうから，これは秘かな屈辱技。連続技に織り交ぜてくるのを読んで中段当て身投げを出せば，ザコは身分をわきまえるようになるだろう。

残念ながら，足払いなどを投げるような下段当て身投げは持っていないので，足払い系統の技には弱くなってしまう。愚かなザコどもは，ギース様が当て身投げを狙うのを読もうとして，足払いを出すことが多い。しかし，ザコの読みに負けるようでは，ギース様を使わせていただく権利などない。弱点は誘いのためにあるようなもの。ザコが愚かしくも足払いを狙ってきたら，こちらも確実な迎撃ができなければならない。

精進すべし。

ザコは寝ていろ！

同じことをやり続けていたら，途端に読まれて，反撃を食らう。ポイントはザコどもの攻撃を読むこと。わざとらしく当て身投げを忘れたようにプレイをして，ザコが苦肉の策として超必殺技を出したところを当て身投げするのも楽しい悪役としての技だろう。

相手の裏をかく技のひとつに，疾風拳がある。ザコどもの頭上から飛び道具が撃てる技だ。この技は反動が大きいので，撃ったあとにふんわりと体が浮いてしまう。うまいところに撃ち込めばいいし，着地点をそらす技として有効ではあるが，多用は禁物，落ち際は弱いのだ。

貴様らザコの考えることはすべてお見通しだ！ そうなるようでなければ，ギース様を使わせていただくのに申し訳ない。対戦はあくまでも読み！である。

特別編7

問答無用の核弾頭ヘッドだ！

Axel Hawk

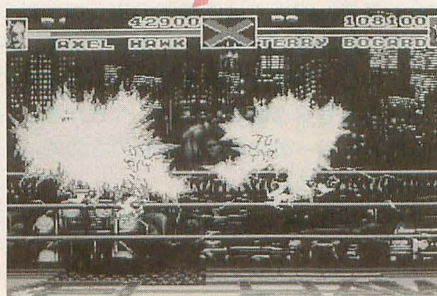
Sudou Yoshimasa 須藤 芳政

そろそろ秋も深まり冬の足音が聞こえてきそうな季節だというのに，持ち前の皮下脂肪を活かしてトランクス一丁で戦う男がいるではないですか。ゲーム中に「ハクソー！」などと，日本人の私たちには誤解を招きかねない発音で登場してしまう彼のトランクスはとてつもなく巨大で，ケンちゃんシリーズで一世を風靡した「デカパンさん」像を彷彿とさせます。ピコピコハンマーで叩いただけで激怒しそうなくらい血の気が多そうな中年オヤジ，アクセル・ホーク(某ハードロックバンドのヴォーカルの名を流用？)はこのたび,餓狼伝説で活躍したマイケル・マックスからトルネードアッパーを伝授されました。ハリケーンやトルネードに比べてタイフーンは人を小バカにしたような「フーン」の響きが気に入らないので，これはよい選択といえます。

さあ，それではいまから皆で地獄のテンカウントを聞いてみよ～！

ブオン！ ブオン！ イェー！

アクセルはキックを使いません。何があろうと使わないのです。缶ケリをやってもわざわざ缶にパンチしてモタつきまくり，いつまで経っても鬼のままなのでしまいにイジケて家に帰ってしまうタイプですね。

しかし，キックが使えない代わりにパンチは強力ですよ。もっとも，キックが使えないうえにパンチがヘロヘロじゃ，しょーもないカスキャラ&肉の塊になってしまいますからこれは当然といえます。しっかし彼は足がノロくてイライラしますね。あまりに歩行がノロいのでムーンウォークでもやってるのかと思いましたよ。

ライン飛ばし攻撃でびりびりびり

さて，初めてアクセルを使う人はアクセルダンス専門家になって，ひたすら跳び込んではパンチボタンを連打しましょう！ この「えっさっさ攻撃」をマスターできれば素人さん相手に連勝間違いなし！ 対戦相手が「つまんないよ～」とボヤいたら，「素人ふぜいが生意気ぬかすな！ これは愛のムチだ！」とでもいいながらダイレクトに地獄突きキャンセルからのニーパットでおとなしくさせましょう。

バスチューアップ！

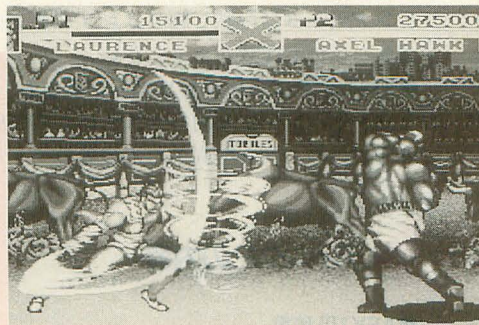
アクセルは跳び込み強パンチ攻撃が強いキャラで，うまく当てると相手に2発の攻撃を加えることができるのですが，これは普通の跳び込み感覚で強パンチを出しても無理です。相手の頭よりも高すぎるぐらいの場所で強パンチを出すと，ガスッ！ ガスッ！ という具合に入るので，これがヒットしたあとに弱パンチからアクセルダンスにつなげば，5発以上相手に連続攻撃を叩き込むことができ，たとえ跳び込みがガードされたとしてもアクセルダンスでかなり削れます。これだけの繰り返しでも対CPU戦ならなんとかなりそうですね。

えっさっさ。えっさっさ……

ばすちゃんや～ん？

こうなってしまうとトルネードアッパーの使い道は，相手の飛び道具打ち消し程度。せっかく伝授したマイケル・マックスの面目丸潰れ。しゃがみガードからトルネードアッパーを出そうとするとスマッシュボンバーになって「えっさっさ！ オウ！」と逆に反撃を食らってしまうのも悲しいところです。まあ，適当に立ち弱パンチやキックからトルネードアッパーにつなげて使ってあげてください。

超必殺技のアクセルラッシュは，ゲージが赤くなったら必ず使わなければならないのが鉄の掟なので使ってください。「そんなこといつ東京サミットで決まったんだ！」などと憤慨のお方には南米アマゾンの吸血ドジョウにでも噛まれてもらうことにして，それ以外の方々は，豪快なアクセルのボクシング魂を学んで「ぼくさぁ，ボクサーなの」のギャグを1日も早くモノにしてくださいね。

避け攻撃は全キャラ中最大のリーチを持つ

トルネードアッパーって役に立たない？

[特別企画]

XL/Imageお試し版+α

XL/Image Alpha

年間5回。
収録しなければならないものがあり，時間的にいささかでも可能性があるときには，すべて付録ディスクをつけたということになる。
そして今回，初めての市販ソフト体験版収録となった。これまでDōGA CGAシステム関係では体験版のようなかたちのものを収録していたものの，意味あいはまったく違っている。機能を限定しても魅力あるツールで，気軽には手を出せない価格帯のものならば，こういう紹介の仕方ももっと増やしていくべきかもしれない。なお，収録プログラムはメーカーの好意によるものなので，くれぐれもパッチ当てなどやらないように。
来年のことについては未定。少しは時間をかけたいところだが……。

付録ディスクの使い方

今回の付録ディスクではOh!Xの付録ディスクとしては初めて市販ソフトの体験版を収録してみました。注目している人も多いソフトですし，値段も高いので興味があってもおいそれとは試せない類のソフトですからこういった体験版にはちょうどよいのではないでしょうか。とりあえず触ってみてください。

こういったツールなら機能制限は多いものの割り切って使えばそれなりの使い方が考えられると思います。

今回の付録ディスクには展開後の容量でディスク3枚分のプログラムとデータが圧縮されています。圧縮ファイルはLHA(LH5)の自己解凍形式になっています。そのまま実行すればカレントディレクトリに所定のファイルとディレクトリを吐き出しますが，ここでは一度フロッピーディスクに展開しておくことをおすすめします。

まず，あらかじめ3枚のフロッピーディスクを用意し，フォーマットしておいてください。次に付録ディスクをドライブ0に入れリセットを押してください。あとは表示されるメッセージにしたがってディスクを入れ替えさえすれば，それぞれのディスクができあがります。

自前で展開作業を行いたいという人は表1のディレクトリ構成を頭において作業してください。

ディスク1の内容

XL/Imageレンダラーのお試し版，およびXL/Image用ツール，各種ツール類を収録しています。

●XL_DEMO

機能制限版のXL/Imageです。ポリゴンしか扱えません。サンプルデータ4つとフルスペックのXL/Imageでレンダリングしたデモ画像を含みます。詳しくは解説記事をご覧ください。

●XL_MODEL　(菊地功)

XL/Imageで使用できる回転体生成プログラムと，書体倶楽部のアウトラインフォントを使用して，指定した文字をXL/Image用のデータファイルに変換するロゴ作成ツールが含まれています。

なお，ロゴ作成ツールで生成されたモデリングデータはお試し版ではレンダリングできないので注意してください。

●X680x0 TeXバージョンアップキット

ソフトバンクより刊行されている『X680x0 TeX』の不具合を直すための差分ファイルです。

ディスク2の内容

●ASK3用辞書

これはASK68K ver.3.0用の辞書を編集部でエディットしたものです。以下の手順で使ってみてください。

1）とりあえず学習なしで使ってみる
2）不都合なものが出たらメモしておく
3）ある程度たまったところで学習ONにしてそれらを変換する
4）その辞書はマスターとして保存する

という手順で使うとよいでしょう。

従来使っていた辞書との差分を加えるときは，

1）自分の辞書とASK3標準辞書の差分を取る
2）差分をASK3辞書に変換する
3）X.DICとの差分を取る
4）差分をASK3辞書に変換する
5）X.DICにマージする

のようにしてください。いきなり差分を取るととんでもないことになります。

大半は思いつくままに足していったものですから，収録語は網羅的ではありません。そのままだとあちこちに穴があるので，一応，国語辞典や各種用語辞典を通読して一般語をピックアップするということもやりましたが，まだまだ抜けているものはたくさんあります。

変換候補の並び順もあまりいじっていません。ベースになっているのはデフォルト辞書ですが，何種類かの辞書ファイルから学習内容を加工したため，少しぐちゃぐちゃになった状態から再構成されています。よって，あちこちにおかしな部分があります。また，学習コードなどは手作業で直してありますが，辞書構造は解析していません。間違ったデータが書き込まれている可

能性もあります。

＊　＊　＊

とまあ，多少問題はありますが，それなりに使いものになるのではないかと思いますのでASK3ユーザーの方は参考にしてみてください。

ディスク3の内容

各種ゲームをまとめてみました。

●WHITE FLAG　　　　(柴田淳)

システムX探偵事務所で連載制作していたピンボールゲームの完成版です。音楽担当は高橋哲史，効果音は西川善司でお送りします。

●PushBon！.SXB/スクロール.SXB　(石田伯仁)

ついに！　という感のあるSX-WINDOW用Push Bon！はSX-BASICで記述されており，独自のピコピコエンジン上で動作します。ピコピコエンジンはゲーム専用の入出力システムで従来のウィンドウエンジンが扱っているビットマップよりも高速に表示を行います。スクロール.SXBも見ていただければだいたいのポテンシャルがわかるのではないでしょうか。

PushBon！.SXBのルールは，従来のPushBon！と同様です。ただし，押せる石がないところでもステップ数が増えるので誤操作に気をつけてください。基本操作は，移動にテンキー，XF2で石を押し，XF1でギブアップとなっています(カラーコーディネート：中野修一)。

スクロール.SXBはスキーゲームです。マウスを左右に動かし，木を避けながら滑って距離を競います。

なお，ピコピコエンジンの機能は将来的に特殊ビットマップのかたちでウィンドウエンジンのアイテムとして取り込まれる予定です。

●SX-BASIC ver.0.6　　　(石上達也)

ディスクにはSX-BASICの最新版も入っています。これはおまけという感じですが，かなりバグ取りされていますのでSX-BASICユーザーの方はこちらを使用してください。今回収録されているゲームはこのバージョンのSX-BASICで動作します(path＄を使っているだけだが)。

また，シャーペンでエディット中のプログラムをSX-BASICで実行するrun.ex(田村健人)も収録しました。

シャーペンのキー定義ファイルに，

M1,'run',　＄0D　　＊　OPT.1　＋　R

のような指定を行ってください。この場合ならOPT.1＋Rで即起動できます。直接メモリハンドルを渡しますのでセーブしなくてもかまいません。

●SHOCK WAVE2　　　(江川乃誉司)

もみじ狩りPRO-68Kで収録したサンプルゲームにリプレイ機能を組み込んだものです。EXEC.FNCを組み込んだX-BASICで立ち上げてください。

起動時にTRACE.DATを読み込みます。サンプルデータがありますので，動作を確認してください。起動画面でジョイスティックを横に入れるとトレースデータの再生が始まるはずです。なお，トレースデータはハイスコアでクリアしたときのみ書き込まれます。前のデータは更新されてしまいますので，残しておきたいデータはリネームして取っておいてください。

●各種懐ゲー集

今月号に掲載されている「なぜかなんとなく懐かしい雰囲気の漂うゲーム」たちのプログラムリストです。詳しくは各記事を参照してください。

＊　＊　＊

毎度のことですが，3.5インチユーザーの方にはメディアコンバートを行っていますので，住所を書いた返信用の封筒と切手260円分を同封して編集室まで送ってきてください。

表1　収録プログラム一覧

```
Listing of Archive : disk1.Lzh

<< dir >> xldemo/
 XLDEMO              .X
 X                   .PPD
 ICE                 .R
 ROTOR               .LPC
 PPD                 .DOC
 PLANE               .PPD
 X1                  .CMD
 X2                  .CMD
 X3                  .CMD
 CHOPPER             .PPD
 sample2             .ice
 sample1             .ice
 README              .DOC

<< dir >> xl_model/
 XL_Rot              .c
 RevLine             .s
 XL_Font             .c
 makefile
 XL_Rot              .x
 XL_Font             .x
 char                .cmd
 rot                 .cmd

<< dir >> tex/bin/
 preview             .x
 print               .x

<< dir >> tex/drivers/doc/
 verup               .doc

<< dir >> tex/fontman/configs/
 texfonts            .def

<< dir >> tex/fontman/doc/drivers/
 readme              .doc

<< dir >> tex/fontman/
 jxl4                .sys
 totex               .sys

<< dir >> tex/mf/bin/
 gftopk              .x

<< dir >> tex/
 INSTALL             .DOC
 TeXup               .doc
 XTeXseigo           .tex
 install             .x

<< dir >> caltab/
 CalTab              .s
 HIOCS               .EQU
 KEEPCHK             .S
 PSPDEF              .H
 CalTab              .mak
 CalTab              .x

Listing of Archive : disk2.lzh

<< dir >> / (root)
 L6                  .DIC


Listing of Archive : disk3.Lzh
<< dir >> shockwave2/
 SHOCK2              .BAS
 trace               .dat

<< dir >> sxbasic/
 SxBasic             .x
 sx                  .c
 comp                .c
 efunc               .c
 inter               .c
 sxbasic             .h
 exp                 .c
 sxbas               .mak
 run                 .c
 run                 .ex
 run                 .mak

<< dir >> sxピコピコ/PushBon!/
 PUSH_BON!           .MAP
 PYOE                .PCM
 THROW               .PCM
 HANSHA              .PCM
 HIT                 .PCM
 sxpushbon           .rec

<< dir >> sxピコピコ/ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ/
 ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ         .Pen
 ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ         .has
 ﾋﾟｺﾋﾟｺｴﾝｼﾞﾝ         .x

<< dir >> sxピコピコ/
 ｽｸﾛｰﾙ               .SXB
 ｽｸﾛｰﾙ               .lb
 PushBon!            .lb
 PushBon!            .SXB

<< dir >> pikopiko/
 hoku                .bas
 BUNNY               .H
 BUNNY               .C
 galax               .c
 galax               .dat
 mcm                 .bas
 mc_circle           .s
 mc_init             .s
 mc_line             .s
 mc_main             .s
 mc_put              .s
 wait                .s
 MAKEFILE

<< dir >> pinball/
 ZMSC                .X

<< dir >> pinball/sources/
 ball                .c
 ballelse            .c
 ballini2            .c
 ballinit            .c
 bumper              .c
 flevent1            .c
 flevent2            .c
 flipper             .c
 flshifts            .c
 idoleplay           .c
 ledmanager          .c
 lights1             .c
 lights2             .c
 macintools          .c
 messages            .c
 minigames           .c
 pbmain              .c
 peinit              .c
 traffic             .c
 ball                .h
 balldefine          .h
 ballext             .h
 gamedefine          .h
 gameglobals         .h
 macindef            .h
 pbdef               .h
 pedit               .h
 peglobals           .h
 x68keys             .h
 ball                .pro
 ballelse            .pro
 ballinit            .pro
 bump                .pro
 bumper              .pro
 flevent1            .pro
 flevent2            .pro
 flipper             .pro
 flshifts            .pro
 ledmanagers         .pro
 lights1             .pro
 lights2             .pro
 macintools          .pro
 mapedit             .pro
 messages            .pro
 minigames           .pro
 peinit              .pro
 pewindows           .pro
 traffic             .pro
 copybits            .s
 spclip              .s
 zmPlay              .s
 makefile
 eventsourc          .h

<< dir >> pinball/
 PB                  .x
 flip                .sp
 ball                .SP
 mask1               .WFD
 mask2               .WFD
 pmap1               .WFD
 pmap2               .WFD
 main_int            .ZMD
 main_loo            .ZMD
 pin_drop            .ZMD
 pin_mg              .ZMD
 sounds              .ZMD
 pinb_pcm            .ZPD
 dai__               .pic
 leds                .pic
 mask1               .pic
 mask2               .pic
 parts               .pic
 pinball             .bat
 HighScore           .WFD
```

[特別企画]
XL/Image
お試し版＋α

使い方と活用の手引き

XL/Imageお試し版について

編集部

**イマジカテクノシステム「XL/Imageレンダラー」のお試し版です。
あくまでも「お試し版」なので実用性には欠けますが，
国内レンダラの最高峰に触れてみてください。**

ディスクの中にはXL/Imageレンダラーの機能縮小版と3種類のサンプルコマンドファイルが収録されています。

とりあえず，

A>XLDEMO

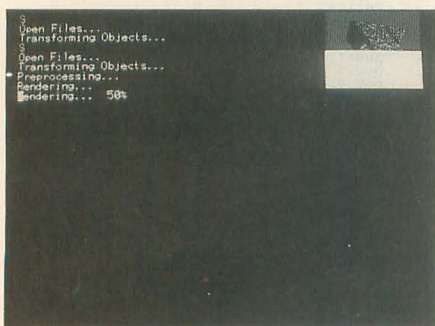
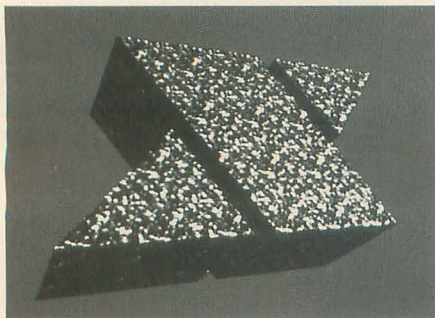
のようにお試し版を起動してみてください。　次に，

<　X1.CMD

のように“<”と空白を1個以上空けてサンプルファイル(拡張子が.CMDのもの)を指定してみてください。ファイルを読み込んでレンダリングが始まります。

そのほか製品版ではどのようなことができるのかというのを示すサンプル画像が256×256ドットで収録されています。なお，このサンプル画像はICE.Rで圧縮されていますので，付属のICE.Rを使って，

ICE SAMPLE1
ICE SAMPLE2

のようにして画面に表示させてご覧ください。なお，このページに掲載されている写真はお試し版に付属するデータをXL/Image(非お試し版)で全画面描画したものです。

レンダリングの様子

サンプルデータその1

その2

製品版との違い

ディスク容量や必要メモリ容量などの関係上，お試し版の機能はかなり制限されたものになっています。

・描画範囲の固定

生成できる画像は128×128ドットの大きさに固定されています。製品版では画像はディスプレイいっぱい，またはファイル上に無制限の大きさでレンダリングできます。

・形状の制限

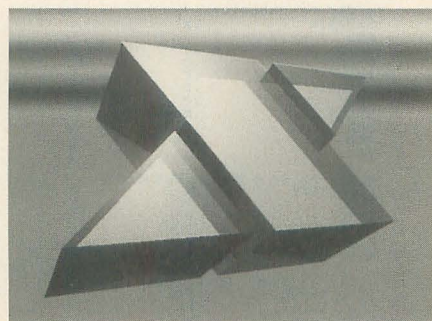
製品版でサポートされている2次曲面によるプリミティブやメタボール，リング，ビームなどが使用できません。扱えるのはポリゴンによる形状ファイルだけです。また，形状間の論理演算ができません。

・形状の使用個数

ポリゴンファイル3つまでに制限されています。

・ファイル出力の制限

LPCデータファイル（RGBαZ40ビットハイカラーデータフォーマット）書き出しには対応していません。画面のみに出力します。

・アンチエイリアス不可

画像のエッジを目立たなくするアンチエイリアスに対応していません。

・ソリッドテクスチャの制限

ソリッドテクスチャといってもピンとこないかもしれませんが，製品版では8種類の質感を表すマッピング素材生成関数が登録されているのに対し，お試し版では2種類しか登録されていません（「WRINKLE（しわしわ）」と「NOISE（ノイズ）」の2種）。

・反射，影落とし不可

レイトレーシングの醍醐味の双璧といえば反射と屈折です。お試し版ではレイの透過，屈折処理には対応していますが，物体に影を落とすことや映り込みの表現ができません。

・マッピング種類の制限

カラーマッピング（いわゆるテクスチャマッピング）とバンプマッピングのみに対応しています。ポストマッピング，環境マッピングには対応していません。

・発光体使用不可

発光体（ファイア型形状指定）のアトリビュートが使用できません。

・光線の減衰不可

距離とともに減衰する光源の指定ができません。

・デプスキュー不可

フォグ処理（空気遠近法）を行うデプスキューが使用できません。

・分散レイトレーシング不可

焦点ぼけやモーションブラー，半影などを実現する分散レイトレーシングに対応していません。

＊　　＊　　＊

以上のようにさまざまな制限があります。製品版と比較するとずいぶん貧弱なものになっているようにも感じられますが，レンダリングサイズやオブジェクト数の制限を除けば，これでも軽くDōGAのREND以上の機能を持っているのがわかると思います。

操作方法

XLDEMO.Xを起動してから有効になるコマンドについては表1を参考にしてください(helpと打つと表示されます)。コマンドは小文字です。エラーが発生したときは必ず,error_initを実行するか再起動してください。

自分でデータを作るには

コマンドファイルを作らなければなりません。なお,ポリゴンデータファイルのフォーマットはドキュメントにまとめられていますので参考にしてください。

コマンドファイルについて詳しく解説するには,クラスがどうのインスタンスがどうのサブクラスが……という話をしなければならないのですが,ややこしくなるのでこれは製品版のマニュアルに譲りましょう。

リスト1にX1.CMDの内容を解説したものを掲載します。だいたいの雰囲気で理解しておいてください。とりあえず,付属のファイルをいろいろいじってみるのがよいでしょう。

パラメータは非常に多彩ですので,このスペースでちゃんと説明するのはまず不可能です。詳しくは製品版を参照してください。試しに,surface defaultで指定できるものをざっと挙げてみましょう。

rgb	G B	色指定
ambient	R G B	環境光強度
diffuse	R G B	拡散率
specular	R G B	反射率
spc1	W U V	ハイライト
spc2	W U V	ハイライト
ph_spec	N	ハイライト
metal	C0 C1	金属感
reflection	R G B	反射率
refraction	R G B	屈折率
opacity	R G B	透明度
filter	R G B	背景反映度
surf_coord	tf_name	異方性反射
color_map	map_name	画像貼付
bump_map	map_name	バンプ指定

これだけではとても理解できないとは思いますが,健闘を祈ります。

マッピングデータを作るには

マッピングで使用するのはLPC(LINKS PIC)という形式のデータです。

この形式はデータのビット数や構成要素,並び順も自由に設定できるなど,非常に柔軟な構造を持っていますので解説は難しいのですが,X68000でもっとも簡単に作成できる形式のものを表2にまとめてみました。ヘッダは1024バイト固定で,先頭からのOFFSETで表内にまとめてあるものだけがXL/Imageで参照されます。

その3。ヘリコプター

角度を変えてレンダリング

あとは512×512ドットの画面に描いたデータを,このヘッダの後ろに分解した色コード(0~31)をGRBAの順で(Aは透明度:0~31),画面の左上隅から走査線方向にスキャンするように指定していってください。

16進数化した数値をASCIIコードで書き込む部分があります。注意してください。

表1

init	... XL/Imageを初期化します
exit	... XL/Imageを終了します
draw [f]	... ワイヤーフレームを描画します(f:高速)
view	... レンダリング描画します
clear R G B	... 画面をクリアします
version	... XL/Imageのバージョンを表示します
error-init	... エラーから回復します
laptime	... レンダリング時間を表示します
sbrk	... システムブレーク値を表示します
sh COMMAND	... シェルを実行します
cls	... テキスト画面をクリアします
< FILENAME	... コマンドファイルを読み込みます
global	... グローバル値を設定します
select	... インスタンスを選択します

表2

Offset	Length		
4	1	format	$00
5	1	n_component	$34
12	4	x_offset	$20202030
16	4	y_offset	$20202030
20	4	x_size	$20323030
24	4	y_size	$20323030
28	5	component	$7267626100
44	5	component_length	$0505050500
60	4	x_proportion	$20202034
64	4	y_proportion	$20202033
256	100	filename	(ASCIIZ)
416	8	file_size	$20203130
			$30343030

リスト1

```
; サンプルコマンドファイル 1
;
;   <バンプマッピング>
;
select  object  obj_root                    物体指定 obj_rootを使用
select  camera  camera                      視点指定 cameraを使用
select  background      background          背景指定 backgroundを使用

create object default obj_root              標準物体の obj_rootを定義
        light   light1  on                  light1 の光源を使用
        light   amb     on                  amb の光源を使用
        object  obj_all                     下位物体 obj_allを使用
close

create  object  default obj_all             標準物体の obj_allを定義
        object          obj_X               下位物体の obj_Xを使用
        rot     -30 30 0                    X,Y,Zで-30,30,0の回転
        mov     0 0 0                       X,Y,Zで0,0,0の移動
close

create  object  default obj_X               標準物体の obj_Xを定義
        geometry        geom_X              形状 geom_Xを使用
        surface         surf_X              質感 surf_Xを使用
close

create  geometry ppd    geom_X              PPDファイルの形状 geom_Xを定義
        file-name       x.ppd               ファイル名 x.ppdを指定
close

create  surface default surf_X              標準サーフェス:surf_Xを定義
        ambient 0 0 0                       環境光の影響 (R G B)
        diffuse 0.3 0.3 0.3                 拡散の指定   (R G B)
        specular 0.8 0.8 0.8                反射の指定   (R G B)
        spc1    0.9 0.1 0.1                 ハイライト指定1 (強度 U V)
        spc2    0.1 0.4 0.4                 ハイライト指定2 (強度 U V)
        metal   1 1                         金属度 (ハイライト 映り込み)
        bump-map map_X                      map_Xをバンプマッピング
close

create  map     solid   map_X               ソリッドマッピングの map_Xを定義
        volume  vol_X                       ソリッドテクスチャ vol_Xを使用
        transform tf_X                      マッピング座標系 tf_Xを使用
close
create  volume  wrinkle vol_X               しわしわマッピングの vol_Xを定義
        magnitude       0.975               ノイズの空間周波数 0.975
        degree          1.75                凹凸の大きさ 1.75
close
create  transform surface tf_X              マッピング座標系 tf_Xを定義
        project solid                       ソリッド座標系使用
close

create light default    light1              標準光源 light1を定義
        type    parallel                    平行光線を使用
        pos     3 0.2 1                     光源の位置
close

create light default    amb                 標準光源 ambを定義
    type ambient                            環境光を使用
close

create camera default camera                標準視点 cameraを定義
        pos     0 0 11                      位置指定
        tar     -0.3 0 0                    ターゲット位置
        perslen 2                           スクリーンとの距離
close

create  volume  background  background      背景 backgoundを定義
        rgb     0 0 0.5                     色指定
close

clear 200 200 200                           画面を指定色でクリア
draw                                        ワイヤーフレーム表示
view                                        レンダリング
exit                                        終了
```

[特別企画]
XL/Image
お試し版＋α

XL/Imageレンダラー製品版対応

XL/Image用モデリングツール2種

Kikuchi Isao 菊地 功

究極の3DレンダラともいえるXL/Image
問題となるのはモデラの不在です
そこで手軽に形状を作るためのツールをお届けします

久しぶりにX680x0で本格レイトレーシングソフト「XL/Image」が発売されます。このソフト，元はUNIX上のアプリケーションだそうで，二次曲面，ポリゴン，メタボールの描画から，マッピングデータの自動生成まで，描画能力としてはほぼ完璧といえる機能を装備しています。

しかし残念なことに，X68000版にはモデラやオブジェクト配置などの支援ツール類がほとんどありません。かろうじてDōGAやZ'sTRIPHONYの形状データのコンバータは付属していますが，基本的に形状データやオブジェクトの配置を記述するコマンドファイルはテキストエディタなどでガリガリ書かなくてはなりません。

これではいかに機能が優れていても，初心者は敬遠しがちになってしまうでしょう。そこで今回は，ポリゴンに絞ったXL/Imageの簡易形状作成ツールを2つほど作ってみました。

XL_ROT.X

ひとつ目は回転体の形状作成ツールです。

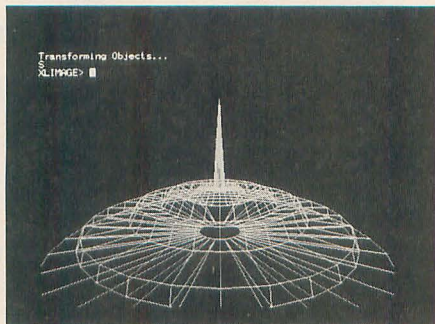
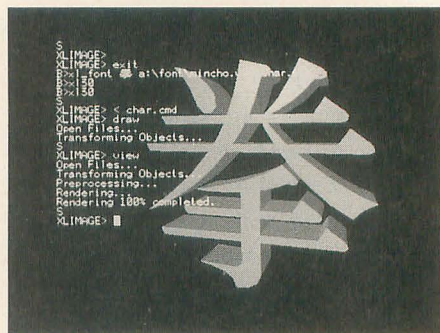
XL_ROT<リターン>

とキーボードから入力して実行してみましょう。すると，画面にマス目が表示されます。左端にある白い実線が回転軸で，XL/ImageではY軸に相当します。ここでマウスで適当なところを左クリックしてみてください。最初にクリックしたところからラバーバンドが伸びましたね。続けて画面右上に表示されている座標を目安にどんどんクリックして断面形状を作成してください。間違った座標をクリックしてしまった場合は，右クリックすることでひとつ前の座標をキャンセルできます。

最後まで指定し終わったら，リターンキーを押してください。すると，今度は回転分割数を聞いてきます。360度の回転をいくつのポリゴンで分割するかを指定するのですが，スムーズシェーディングを行うこともできますので，むやみに大きな値を指定するのはやめましょう。

値を入力すると，ポリゴン形状データであるPPDファイルの内容が標準出力に出力されますので，ROT.PPDというファイルに落としたければ，起動時に，

XL_ROT ＞ ROT.PPD

のようにリダイレクトしてやってください(手抜きです)。さて，これでPPDファイルができましたので，さっそく描画させてみましょう。このROT.PPDを表示するためのサンプルコマンドファイルROT.CMDを用意しておきましたので，ROT.CMD中のPPDファイル名を自分が作成したファイル名に変更してからXL/Imageを起動してください。今月号の付録ディスクのお試し版であればXLDEMO.X，製品版であればXLIMAGE.XまたはXLIM030.Xです。すると，

XLIMAGE>

というプロンプトが現れるはずです。そこで，

XLIMAGE> ＜ ROT.CMD
XLIMAGE> VIEW

と続けて入力してください。

うまく表示されたでしょうか。もしエラーが発生した場合は，ファイルがカレントにないと考えられますので，ファイル名をフルパスで記述するか，

XLIMAGE> EXIT

と入力していったん終了してから，コマンドファイルとPPDファイルのあるディレクトリに移動してから，再び起動してください。

さて，気にいった形状ができあがったら，今度は自分でコマンドファイルを書くことになると思いますが，その際に注意があります。XL/Imageでは頂点指定が左回りに見えるほうが表，反対が裏であると認識し，デフォルトでポリゴンの裏は描画しません。XL_ROT.Xで生成されたポリゴンデータは，画面の上から下に向かって指定された断面では回転の外側が表，内側が裏，下から上に向かって指定された断面ではその逆になります。どちらの面を描画するかはオブジェクトクラスのsideで変更できますので，状況に応じて指定してください。

文字をポリゴンに

適当に線を引いて

回転させるとこうなる

XL_FONT.X

ツァイトの書体倶楽部のフォントをポリゴンで形成されたソリッドモデルに変換します。書体倶楽部のフォーマットについては本誌1992年6月号に掲載されていますのでここでは説明しませんが，問題はXL/Imageのポリゴンが凸多角形にしか対応していないことです。「ここまでやっておいて，なんでこんなお粗末な仕様なんだ？」と思い，凹多角形を描かせてみたところ，ちゃんと描画される部分もあるものの，確かに正常には描画されませんでした。文字面を三角形分割するという方法もあったのですが，かなり面倒そうだったので，ブーリアン演算を行うことにしました。

ということで，残念ながら今月号のお試し版ではXL_FONT.Xで生成されたデータを描画することはできません（だって体験版がブーリアン演算できないの知らなかったんだもん，いまさら作り直すのやだし)。

お試し版でXL/Imageが気にいった方は（あれくらいでは全貌はわからないとは思うが）製品版を買って，このツールを試してみてください。

ただし当然，書体倶楽部のフォントを持っていない方も利用できません（Z's STAFFに付属のフォントは利用できます)。

XL/Imageでは，ポリゴンの裏と裏で挟まれた領域を「中」と認識しているようです。そこで，試しにポリゴンで作った側面だけの筒Aと直方体Bの論理積を取ったところ，図1のように円柱Cの底面に穴が開いてしまいました。

視線を伸ばしても筒Aと交点を持たない部分は認識できないようです。この現象を避けるには，筒Aを長く取ってやればいいのですが，十分大きな値を指定すると，誤差が生じたりオーバーフローして正しく描画できなくなってしまいます。しかたがないので，図2のようにして筒の両端を閉じてやることにしました。面数は多くなってしまいますが，こうすることで完全に立体を閉じることができますし，図3のようにすれば穴の開いた形状も作成できました。

これでだいたいのめどが立ったと思ったのですが，もうひとつ問題がありました。書体倶楽部のフォントの輪郭も左回りだと思い込んで，しかもテストに使っていたフォントがたまたま左回りだったので，私もあとになって気がつきましたが，回り方向が決まっていないのです。回り方向が逆になると，側面も表裏逆になってしまいますので，これは致命的です。そこで，X軸に平行な直線を考え，それと交差するベクトル群の順番と向きから，それぞれのベクトルが属する閉じた領域がどちら回りなのかを識別しました。

ただし（ベクトルフォントの性質からいってないとは思いますが）輪郭が交差していたり，まったく重なっていたりした場合には正しく変換できません。

それでは実際に変換してみましょう。使用法は，

XL_FONT ＜Char＞ ＜Fontname＞ ＜PPDname＞

です。

＜Fontname＞で指定した書体倶楽部ベクトルフォントファイルの文字＜Char＞を，XL/Imageのポリゴン形状データファイル＜PPDname＞に変換します。＜Char＞は半角でも全角でも構いませんが，最初の1文字を対象とします。また，フォントファイルには拡張子が“VF1”の第一水準と“VF2”の第二水準がありますが，プログラムで識別して拡張子をつけますので，＜Fontname＞には拡張子をつけなくて構いません（つけても無視されます)。

XL_FONT 鬱 GOTHIC CHAR.PPD

てな具合ですね。

フォントファイルがカレントにないときには，パスをつけて指定してください。

すると，CHAR.PPDというファイルができますので，さっそく描画させてみましょう。こちらもサンプルコマンドファイルCHAR.CMDを用意しておきましたので，さっきの要領で，

XLIMAGE＞ ＜ CHAR.CMD

XLIMAGE＞ VIEW

と入力してみましょう。

描画できましたね。

側面にスムーズシェーディングをかけたい場合はCHAR.CMDの38行目の先頭の';'を削除して再描画させてみましょう。そのほかにもコメントを入れておきましたので，各自でいろいろいじってみてください。ただし，ブーリアン演算を行いますので，global csgは必ずonに，側面オブジェクトのsideはbothにしてください。どうやら描画面がこちらを向いていないポリゴンは計算前にはじかれてしまうようで，ブーリアン演算も行ってくれないのです。あと，大きさやパート分けは図4のようになっていますので，マッピングなどを行うときには注意してください。

最後に

このXL/Image，機能だけでなく描画も結構高速です。コマンドファイルなどを書いておいてリダイレクトしてやれば，子プロセスとして呼ぶこともできそうです。いろいろ面白いことできそうだなぁ。でもエラーはちゃんと出してほしかったところ。csgをcgsと書き間違えてしまって，しばらく悩んでしまいましたよ，私ゃ。

図1 閉じていない立体との理論積

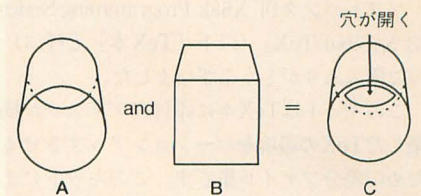



図2 両端を閉じた筒

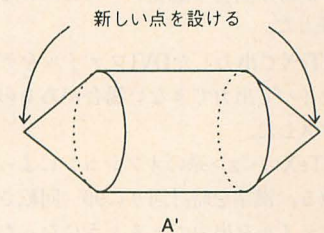
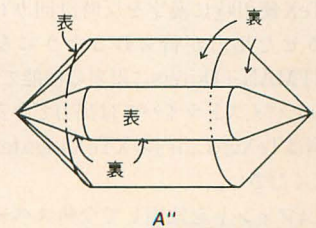
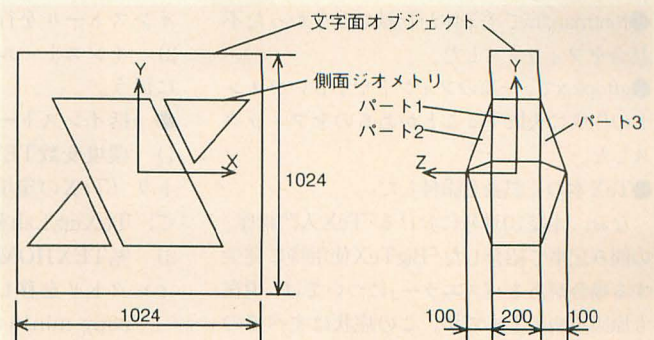



図3 穴の開いた形状作成時の側面構成

図4 各寸法とパート構成

[特別企画]
XL/Image
お試し版＋α

+Alpha

最新版差分収録

X680x0TeX Version Up Kit

ハードウェア活用書編集部

ソフトバンク刊『X680x0TeX』の収録プログラム群のバージョンアップ情報です。
どうしてもインストールできなかった人は参考にしてください。
あわせてTeXをめぐる話題などにも触れてみます。

「はじめに」あるいは「変更履歴」

ソフトバンク刊『X68k Programming Series #3 X680x0 TeX』(以下「TeX本」と呼ぶ)のご購入ありがとうございました。

このキットはTeX本に添付のディスクが構築したTeXの環境をバージョンアップさせるための差分ファイル集です。このキットによる具体的な変更点は以下のとおりです。

●インストーラと書籍本文の不整合とインストーラが行う問答の論理レベルにおける矛盾を修正した。

●pLaTeXで出力したDVIファイルをデバイスドライバで出力できない場合があるのをフィックスした。

●pLaTeXの<z>系のオプションによって出力される，漢字を時計回りに90°回転させたDVIファイルを出力できるようになった(これによってpLaTeXのすべての出力に対応したことになる)。

●LaTeX使用時に漢字を反時計回りに90°回転させた出力が得られるようになった(FONTMAN＋Driversに限定の機能であり，ほかのデバイスドライバでは出力できない)。使用例はTeXup.Lzh内のXTeXseigo.texを参照してください。

●JXL4フォントを使用して全角スペースを含んだソースが出力できなかった不具合をフィックスした。

●fontman.fmで毛筆体が登録できなかった不具合をフィックスした。

●gftopk.xで，一部のフォントや小さいフォントの作成に失敗することがあるのをフィックスした。

●TeX本の正誤表を添付した。

なお，本誌10月号における『TeX入門講座』の囲み記事で紹介した「BigTeX使用時に発生する場合があるバスエラー」については，現在も原因不明のままです。この症状はすべてのX680x0とその環境との組み合わせにおいて発生するわけでも，すべてのTeXソースに対して発生するわけでもありませんから，一度は実行してみることをおすすめします。

ちなみに，TeX本に添付されていたソース中でバスエラーが起こる可能性があるのは，%TEXHOME%¥doc¥jtexdoc.texの7ページめを出力したところ(画面上に「[7]」を出力した直後)です。

また，仮にバスエラーの症状が出てしまう場合でも，BigTeXが実行時に使用するメモリ領域を変更してやると問題なく処理できるようになることが多いようです。具体的には，バスエラーで止まってしまう場合，以下に示す方法でこの症状を回避することができることがあるので試してみてください。

●BigTeXの実行前にCOMMAND.Xを数回(6回くらい?)実行しておく。

●BigTeXの実行より先にFONTMANを常駐させておく(FONTMANの常駐時にバスエラーが出る場合には，常駐を解除してからBigTeXを実行する)。

バージョンアップの方法

本キットによってTeX本の環境をバージョンアップさせるには，以下の手順で操作してください。

●まだインストールをしていない場合

1) TeX本に添付されているinstall.xではなく，TeXup.Lzh内のinstall.xを使用してインストールを行う。

2) インストール方法に応じて以下の手順に従う。

●一括インストールをしている場合

1) 環境変数TEXHOMEが示すディレクトリ(TeXの環境があるディレクトリ)上で，TeXup.Lzhを展開する。

2) %TEXHOME%¥fontsにカレントディレクトリを移し，

copy min10.tfm ytmin10.tfm

を実行する。

3) %TEXHOME%¥preview.p3mファイルに，

ytmin -tex-高速-縦-明朝

を，%TEXHOME%¥print.p3mファイルに，

ytmin -tex-縦-明朝

をそれぞれ追加する。

4) LaTeXでXTeXseigo.texを処理し，出力ないし閲覧する。ただし，縦書き用のフォントが使用可能なコンフィギュレーションでFONTMANを常駐させておかなければならない。たとえば，%TEXHOME%¥fontman¥fontman.fmを親とする一連のファイル(詳しくはfontman.fm内に記された指示に従ってください)を縦書き可能な設定に編集し，これをコンフィギュレーションファイルにしてFONTMANを常駐させておく。

●分割インストールをしている場合

1) 環境変数TEXHOMEが示すディレクトリ(TeXの環境があるディレクトリ)上で，TeXup.Lzhを展開する。

2) 作成された%TEXHOME%¥mf¥bin¥gftopk.xを環境変数MFHOMEが示すディレクトリ(METAFONTの環境があるディレクトリ)の配下にあるbinディレクトリ(%MFHOME%¥bin)にムーブする。

3) TeXup.Lzhを展開したときに作成された%TEXHOME%¥mf¥binは不要なので削除してもかまわない。

4) 「一括インストールをしている場合」の「2」以下を実行する。

以上で，差分の適用は終了です。

X680x0 TeX Q&A

編集部に寄せられた読者ハガキやお手紙の中から，いくつか代表的なご質問に答えてみたいと思います。

Q. 対応しているprint.cfgがない。

A. キヤノンのBJC-600J，エプソンのMJ-

400やMJ-700V2Cなど，ESC/P(Ver.2)系のコントロールコードを持っている360dpiのプリンタの場合に共用できるprint.cfgを紹介します。なお，Offset系のオプションや-prBufSizeは好みと使用機種とに応じて任意に設定してください。もし，ひどく印字が乱れて動作しない場合には，次の質問を参照してください。

```
-remark=360dpi_Printer(ESC/P_Ver.2)
-remark=modified_by_紫苑
-width=2880
-height=3960
-dpi=360
-MSBisUpper
-TRAM
-pinBytes=6
-init=¥e@¥x18¥e3¥x18
-CRLF=¥r¥n
-FF=¥f
-graphic=¥e＊¥x48%2i
-start=
-repeat=
-relative=¥e¥¥%2/2i
```

Q. 該当するprint.cfgがあるのに出力ができない，あるいは出力が乱れる。

A. この場合の原因はいくつか考えることができますので，以下に示す解決策を順に試してみてください。

1) プリンタのコントロールコードが異なっている

使用するプリンタが複数のコントロールコードに対応している場合，print.cfgが想定しているコントロールコードとプリンタで設定しているコントロールコードが異なっていることがあります。プリンタのマニュアル等を参照して，プリンタの制御コードの設定を確認してください。

2) プリンタドライバがオーバーランしている

まず，シャープが提供する汎用のプリンタドライバPRNDRV?.SYSを，どれでもかまいませんからCONFIG.SYSで登録します。次に，print.x実行時にオプション“-dump=lpt”を加えてみてください。このとき，うまく出力されるようになった場合には，プリンタドライバのオーバーランが原因です。

以上の手段で直接lptデバイスにダンプするか，いったんファイルに落とした(print.x実行時に“-dump=output”を指定)あとでlptデバイスにコピーする（copy output lpt）か，どちらかにしましょう。

ただし，前者は1ページ出力するたびに「プリンタがつながっていない」旨の白窓が出るので，そのページが排紙されるごとに「〈R〉再実行」を指示しなければならず，後者は相当にディスクスペースを食います。なお，後者が必要とするディスク容量は尋常ではないので，よほど余裕がないかぎり，前者の方法をすすめます。

3) プリンタのコントロールコードの仕様を越えるような情報を送っている。

TeX本Vol1,210ページの“-relative”オプションの例のように，プリンタのコントロールコードの仕様を越えるような情報を送った場合，印字が乱れることがあります。TeX本の例を参考に，“-dump”による出力とコントロールコードのマニュアルと対比しながら調べてみてください。

4) 使用中のprint.cfgに不備がある。

環境の違いやロットの違いなどで，動作しない場合がないとはいいきれません。プリンタのコントロールコードのマニュアルを参照しながら，print.cfgを確認してください。

Q. XVIでプレビューができない。

A. XVIの初期ロットではpreview.xが動作しないことがあるようです。その場合には，動作クロックを16MHzから10MHzに変更してみてください。

Q. Windowsなどで使用されているTrue Type Fontを使用することはできないか。

A. 日本語TrueType Fontフォントについては，TrueType Fontを使うためのFONTMAN用のフォントドライバが作成されていませんので，現状ではTrueType Fontそのものを使用することはできません。しかし，NIFTY-ServeのFLABOには，TrueType FontをツァイトのJGフォントファイルフォーマットに変更するツール(DOS用)などが登録されていますので，これを使用すれば利用できないわけではありません。英語フォントなどもTrueType Font化すれば，ある程度，フォントのディスク占有率を低下させることができるのですが，そのためにはかなり大規模なデバイスドライバの変更が必要になると思われます。

Q. TeXのマクロ集や周辺ツール集を発売しないのか。

A.……。黙秘権（企業秘密ともいう）を行使します。

Q. LaTeX3はいつリリースされるか。

A. 当分リリースされないと思われます。その理由は，TeX本のVol.,141ページのコラムで触れているLaTeX3のプロトタイプマクロ集がLaTeX2ε(「らてっくツーイー」と読むようです）として正式公開されたからです。欧米ではすでにこのマクロ集が相当普及しているようですが，日本ではどうなるかわかりません。なお，このマクロ集自体はpTeXでも動作させることができますし，PC-VANのSSCIENCEではこの日本語化をされている方がいますので，興味のある人は覗いてみるとよいでしょう。ちなみに，LaTeX2εのマニュアルと手引の邦訳は，某社(A社ではない)から出版されるようです。

Q. 今後のTeXはどうなるか。

A. 先日，TUG（TeX Users Group） 15th Annual Meetingに参加された方のお話を聞く機会がありました。それによると，これまでTeXを引っ張ってきた先人たちの関心は，すでにTeXそのものではなく，次世代へ向けてのpost TeXシステムのプロジェクトに向いていて，その実現のための試みもいくつかなされているようです。実際，LaTeXの作者であるL.Lamport氏も「LaTeX4はLaTeXではない」という発表をやったそうです。また，日本でも『LaTeXによる美文書作成入門』の著者である奥村晴彦さんらを中心として，TeX3を母体としたpost pTeXの動きがあるようです。

なんにせよTeXの周辺は非常に混沌として面白くなってきたので，興味のある人は追跡してみるとよいでしょう。

TeXの参考書

TeX本以外の参考文献を示してほしいという要望がいくつかあったので，紹介しておこうと思う。初心者におすすめなのは，『やさしいLaTeX のはじめかた』(すずきひろのぶ氏)である。もし，この本でLaTeXの最小限の使い方が理解できなくても，事実上，これ以上やさしく書かれた文献は存在しないだろう。

ある程度LaTeXの使い方がわかってきたら，『LaTeXトータルガイド』（伊藤和人氏）や『文書処理システムLaTeX』(L.Lamport氏)あたりで血肉をつけるとよいと思う。

また，初心者から中級者にかけておすすめなのが，『てくてくTeX』（阿瀬はる美氏）である。「はるみちゃん」と「亀之助」の掛合いで進行していく内容で，10月末に2分冊の1冊目が刊行された。内容が詰まっているわりに口あたりがよいので初心者から脱したくなったら，これを読んでみるとよいかもしれない。

いずれにせよ，究極のガイドはjlplain.texやlatex.texである。これらを完全に読みこなしたなら，あなたは立派な〝TeX Wizard〟となっているだろう。

[特別企画]
XL/Image
お試し版+α

RSDRV不要，SX-WINDOW対応

タブレットドライバ改良版

Kikuchi Isao 菊地 功

メーカーとの共同制作によって格段に安定したタブレットドライバです
製品全機種に対応し，RSDRV.SYSも不要になりました
SX-WINDOWやWP.Xでも使用できます

10月号のもみじ狩りPRO-68KにNS CalComp製のタブレットをマウスエミュレーションするドライバを掲載しましたが，ありがたいことにCalCompの方にも見ていただけたようで，私が動作チェックできなかったDrawingBoard-IIやDrawingPadでのチェックをしていただきました。その結果，上の2機種はおろか，DrainwgSlateの最近のものでも動作しないことが判明しました。どうもすみませんでした。

今回のバージョン1.10は，CalCompの方に新機種を含めたその他のタブレットに対応してもらいさらに，読者からいただいた情報をもとにSX-WINDOWにも対応（相対座標のみ）させました。

対応機種

今回のバージョンで以下の機種が新しく対応しました。

DrawingBoard-II
DrawingPad
新DrawingSlate
DrawingBoard-III

新DrawingSlateとは，1994年6月から出荷されたものであり，デザインが一部変更され，ISO認定によりファーム番号が変更されたそうです。

新機能

タブレットの領域を画面全体に対応させると，DrawingSlate31090などの横長の機種ではどのグラフィックモードでもほぼトレース入力ができますが，領域が正方形の機種ではアスペクト比が狂ってしまいます。そこで，今回のバージョンでは図1のように，メニュー13をピックするとタブレット認識範囲が4：3に，14をピックするとディスプレイの描画範囲が1：1になるように変更されました（この新機能はCalCompの方につけていただきました。渡辺さん，ありがとうございました）。

前回のバージョンではWP，SX-WINDOWやソフトウェアキーボードには対応していませんでしたが，読者の小松ひでさんから対応方法を教えていただきました。詳しい説明は省きますが，マウスからのデータをIOCSが加工する前に奪ってしまって，代わりのデータを渡してやる方法です。

しかしマウスからのデータは移動量（とボタン）のデータなので，どうしても相対座標になってしまいます。しかたがないので前回の方法は絶対座標モードとして残し，相対座標モードをつけることにしました。起動時は絶対座標モードですが，メニュー1をピックすることで絶対/相対座標が切り替わります。

これで相対座標モードならばSX-WINDOWやWP.Xで動作するようになります（ソフトウェアキーボードは絶対座標でも動作します）。移動速度の調整はメニューの15～17で行うこともできますし（それぞれ領域全体を256×256，512×512，768×512として相対量を渡している），起動時のオプション/msも有効です。ただし，ちょっと問題なのは，X680x0のマウス（正確には移動量から座標を算出するIOCSなど）は加速度マウスなので，タブレットからのデータも加速度変換されてしまいます。そのため，相対座標モードではトレース入力は難しいかもしれません。

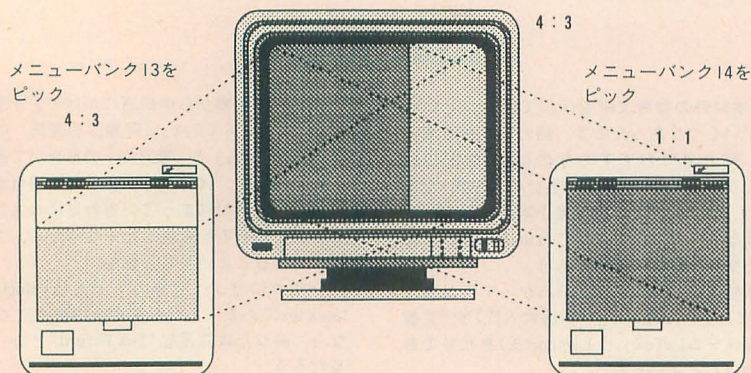
その他，安定性や速度も若干上がっているはずです。

メーカー公認？

このタブレットドライバ，新しくCalCompで出荷されるタブレットには添付されるそうで，この記事が載るころにはお店に並んでいる製品には添付されているかもしれません。すでに製品を持っていて，どうしてもそれがほしいという方は，いますぐCalCompに連絡してみましょう。ちゃんとサポートしてくれるはずです。

今回のドライバのバージョンアップはメーカーとの共同開発のようなものでした。私とCalCompの方が交互に手を加え，締め切りぎりぎりになってRSDRV.SYSを組み込まなくても動作するバージョンができあがりました。

CalCompの方に聞いたところ，同社のタブレットのユーザーでいちばん多いのはX680x0ユーザーだそうです。ちょっと嬉しくなっちゃいますよね。

図1

Oh!X 周年特別企画

懐ゲー制作工房

パソコンでゲームを遊ぶ＝プログラムを打ち込む
そんな図式が当たり前だった時代
パソコンユーザーは，ゲームを遊ぶためだけに
テープでBASICを読み込み，ひたすらプログラムを打ち込まなければならなかった
突然の停電に泣いたり，入力ミスによってプログラムが動かなかったり……
たかがプログラムの打ち込みにも，ずいぶんいろいろなドラマがあったものだ
*
そんなマイコン全盛期に生み出されたピコピコゲームたち
文字キャラクタだけの貧相な画面写真を見ると，とても面白そうだとは思えない
しかし，当時のユーザーたちはそのピコピコゲームを夢中になって遊んだ
なぜなら，文字キャラクタには顔があり，いろいろなアイデアが存在していたからだ
表現力が乏しいぶん，逆にアイデアがゲームの面白さを決めていたといってもいい
まさに，ゲームの本質的な面白さのみで勝負していた
そして，遊び倒したユーザーは自分の手でゲームを作るという欲望にとりつかれ……
*
ピコピコと動くキャラクターを眺めながら，
ちょっとノスタルジックな気分に浸ってみませんか

CONTENTS

ああ，懐かしのゲーム喫茶

ブロック崩し&バトルテニス

Asakura Yuji 朝倉 祐二

トップバッターは朝倉氏が作る懐ゲーの基本中の基本，ブロック崩し&バトルテニス
懐かしのゲーム喫茶を思い浮かべながら遊んでみてください
なお，実行するには10月号の付録ディスクに収録されたXSPRITE.FNCが必要です

懐ゲーと聞いて，まっさきに思い出すのが「ブロック崩し」と「バトルテニス」。その昔，ピコピコという音を奏でながら喫茶店のテーブルとして大活躍してたのを，読者の皆さんは覚えていますでしょうか。マイコン全盛時代にも，構造がシンプルで作りやすかったため，あちこちのBASICでプログラミング入門の題材として見かけましたね。

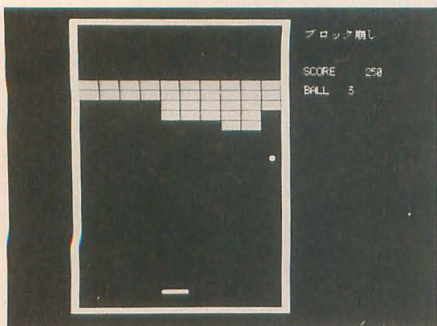
ここまで引っぱればおわかりのとおり，ここでは，ブロック崩しとバトルテニスを制作します。定番といわれようとも「やはり懐ゲーの基本はブロック崩しにあり」という強引な持論により，反論は受けつけません。

そして，「さあ，やるか」と気合を入れると，どうせだから10月号の付録ディスク「もみじ狩りPRO-68K」に収録されたXSPRITE.FNCを利用してみよう，と天の声(悪魔のささやき？）が聞こえてきたのです。スプライトゲームを作るために便利な関数がいっぱいあるし，あっても困らないだろうと簡単に考え，制作を開始しました。

ゲームの説明

まずは，遊び方を説明します。

●ブロック崩しの遊び方（リスト1）

ジョイスティックのボタンを押してゲームスタート。パドルを左右に動かしてボールを跳ね返し，ボールをブロックに当てます。ボールが当たったブロックは画面から消えます。すべてのブロックを消すのが目的です。パドルでボールを3回受け損なうとゲームオーバーです。

ブロック崩し

●バトルテニスの遊び方（リスト2）

CPUとの対戦専用です。ジョイスティックのボタンを押してゲームスタート。プレイヤーは左側のパドルを上下に動かしてボールをパドルに当てて跳ね返します。パドルにボールを当て損なうと相手に1ポイント入ります。どちらかが7ポイント先取するとゲーム終了です。

どちらのゲームもできるだけコンパイルして遊んでください。もちろん，XSPRITE.FNCをX-BASICに組み込んでおくようにね。なお，時間的な関係上，今回の付録ディスクにはソースリストを収録することができませんでした。がんばって打ち込んでください。

ブロック崩しの制作

まず最初にキャラクターをPCGに定義することから始めました。キャラクターといってもパドルとボールとブロックの3つです。形も単純でベタ塗りなので，パターンエディタを使うまでもなく作成できました。パドルとブロックはスプライト1個に収まらないので，sp_hgadd()で横に連結して疑似的に32×16のスプライトとして扱うようにしています。

次に画面デザインを決めます。表示画面を512×512にして画面のだいたい真ん中に長方形のゲームフィールドを描き，その中にバランスよくブロックを横1列10個のものを5段とパドルを配置します。静止画だけ見たらずいぶんとブロック崩しらしくなってきました。

いよいよボールを動かします。ボールが動いたら，ジョイスティックでパドルを動かせるようにしてボールを打ち返せるようにします。ボールが壁，パドルにぶつかればボールの跳ね返り処理を行い，ブロックに当たったのならブロックを消し，跳ね返り処理を行います。これら一連のヒットチェック処理を追加すればゲームはほぼ完成です。

今回はヒットチェックおよびパドルの移動をXSPRITE.FNCで管理しようと，制作開始当初より決めていました。ボールの移動はベクトルの大きさで移動量を決定するsp_slidev()で制御できます。ボールの跳ね返りは移動ベクトルの大きさを変化させることで表現できるので，これも簡単でしょう。XSPRITE.FNCのサンプルプログラムも，敵弾の移動はsp_slidev()を使っていましたね。パドルの移動はsp_stkon()を使うことにします。

ヒットチェックがうまくいかない

ヒットチェックはボールとブロック，ボールと壁のように，必ずボールと“なにか”の衝突判定です。ですから，その“なにか”を接触判定グループ0にまとめて登録しておき，ボールを定義したスプライト番号(リストではSP0）と接触判定グループ0の間でヒットチェックをすることにします。実際はSP0にボールを表示し，SP1とSP2を連結してパドル，周りの壁をSP3～7，ブロックをSP8～107に定義しました。これで，

```
sp_slon(0)
sp_hiton(0,0)
sp_slidev(0,64,-96,600)
s=sp_hit(0)
```

とすれば，sにSP0に接触したスプライト番号を得ることができるはずです。

机上の計算ではうまくいくはずだったのですが，実際にプログラムを書いてみると，ボールが壁，ブロック，パドルにめり込むという症状が出てしまいました。

これはインタプリタの処理速度の遅さが原因と思い，コンパイルしてみたのですが状況は改善されません。sp_hitrng()でヒットチェックの範囲設定を間違えたのかと見直しても，おかしい部分はありません。細かくヒットチェックの範囲を試行錯誤して調整してみたのですが，どうやってもボールがめり込んでしまい，納得のいくヒットチェックがなかなかできません。結局原因がわからず，ブロック崩しの作成も中断を強いられることになりました。

見えてきた問題点

10月中旬に編集室にお邪魔して，Oh!X11月号を読んでいると，XSPRITE.FNCの作者，伊藤氏による「XSPRITE.FNC使用ガイド」が掲載されていました。これを帰宅途中の電車の中で読んでいて，原因がなんとなくわかりました。sp_slidev()の説明のところで，スプライトの「スピード値は32が基準」とあります。つまりスピード値が32だと，1回の垂直帰線期間割り込みで1ドットスプライトを移動するということです。スピードを64にすれば2ドット，96にすれば3ドット，1回の割り込みで移動させることを表します。

sp_slidev(0,128,0,50)

このようにすると，SP0が1回の割り込みで4ドット右に移動します。つまり表示位置が固定しているスプライトと衝突判定をするとき，ある時点で衝突していないスプライトが次の割り込みで最悪3ドットの重なりが起きることになります。

XSPRITE.FNCは割り込み期間中にsp_slidep(), sp_slidev()で指定した全スプライトの移動位置を計算し，スプライトの表示位置を更新したあとヒットチェックをします。ヒットチェックを行うスプライトが双方とも移動していて，スプライトの重なりがそれほど気にならないシューティングゲームならこれでもいいでしょうが，固定の障害物，たとえばピンボールゲームのボールが台上のあらゆる物体にめり込んで衝突判定がされるような状態だと利用価値が薄れてしまいます。

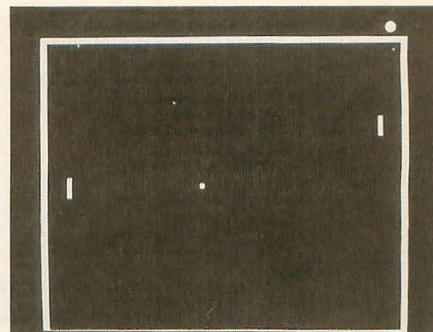
ボールの移動にsp_slidev()を使うかぎり，ボールの移動量が大きくなればなるほど，壁やブロックにボールがめり込んでしまうのは避け難い現象だということがわかって困ってしまいました。だからといって目標であったXSPRITE.FNCを利用したプログラム作成を簡単にあきらめたくありません。

結局，ボールの表示をsp_slidep()で行うことにしました。sp_slidep()はボールの移動先を直接座標で指定するスライド関数です。ボールが壁に触れる座標は左側の壁はX座標＝140，右側の壁はX座標＝385，上側の壁はY座標＝21と固定です。つまり，ここでボールの移動軌跡が描く直線の傾きがわかれば直線を表す1次方程式，

$$y=a(x-bx)+by$$

によりボールと壁の交点を求めることができます。ここでaは直線の傾き，bx,byはボールを表示しているスプライトの座標，x,yにはボールが壁に触れるx,y座標のうちわかっているものを代入します。

こうして壁とボールとの交点を求めたら，sp_slidep()でボールを表示するスプライトを表示します。ボールが壁に接触したのを調べるには，sp_xstat(0)の戻り値を調べて，下位1ビットが0であるならボールの移動は完了しています。その時点でのスプライトの表示座標をsp_statで調べれば，ボールがどの壁に接触しているか判断することができます。

ボールとパドルのヒットチェックもボールがパドルを突き抜けているように見えます。ですから，これもsp_slidep()でボールがパドルと接するY座標までスプライトを移動させるようにして，sp_xstat(0)でボールの移動が終わったことを確認してから，パドルとボールのヒットチェックをするように変更しました。この変更にともない従来SP0と接触判定グループ0の間で行っていたヒットチェックを，SP0と接触判定グループ1，SP1と接触判定グループ0の2つに分けました。接触判定グループ0にボール，接触判定グループ1にブロックを定義してあります。

バトルテニス

ボールとブロックのヒットチェックはこれまでどおりsp_hit()を使いますので，ボールとブロックが重なってからヒットチェックが検出される場合がほとんどです。そのためボールがブロックにめり込んでしまい，ブロックへの当たり方によっては，複数のブロックが一度にたくさん消えてしまうことがあります。なんとかして直そうとしたのですが，いいアイデアが浮かばずそのままになってしまいました。

最後に

XSPRITE.FNCを実際に使ってみて，ブロック崩しのような固定物体との衝突判定が多いゲームの作成には，不向きのような気がしました。こういったツールは実際に使ってみないと，なかなか不具合というものは見つけにくいものです。

XSPRITE.FNCが発表されて間もないため，これからもこういった問題点が出てくると思います。そのときは，あくまでもXSPRITE.FNCはサポートツールだ，と割り切って考えることで解決方法が見えてくることでしょう（今回の場合も解決できない問題ではないはずです）。

それでは読者の皆さん，XSPRITE.FNCを利用したゲームを作成したなら，ぜひとも編集室に投稿してきてください。楽しみに待っています。

リスト1

```
 1: /*
 2: /* ブロック崩し
 3: /*
 4: str a
 5: int i,j,bx,by,blx,bly
 6: int ox,oy
 7: int sc
 8: char speed
 9: char flg=0
10: char svc_flg=1 /* service flag
11: char ball=3
12: char cnt=0
13: float dx,dy,ddx,old_ddx
14: /*
15: screen 1,3,1,1:console 0,31,0:sp_init():sp_xinit():defpat()
16: sp_color(0,0,1):sp_color(10,24151,1):sp_color(15,65535,1)
17: /*
18: locate 51,1:print "ブロック崩し"
19: locate 51,5:print "SCORE"
20: locate 51,7:print "BALL "
21: sp_hitrng(0,3,3,12,12):sp_hitrng(1,0,5,31,10)
22: /*=== パドルを表示するSP1,SP2を横に連結 32*16 ===*/
23: sp_hang(1,2,16,0)
24: /*=== 接触判定グループ0にボールをセット ===*/
25: sp_hgadd(0,0)
26: /*
27: fill(144,0,364,511,1)
28: for i=0 to 8
29:    box(128-i,8-i,380+i,503+i,65535)
30: next
```

```
 31: /*=== パドル表示 ===*/
 32: sp_loc(1,252,480):sp_stkon(1,1,4,0)
 33: while 1
 34: display_block():cnt=0:sc=0:ball=3
 35: locate 56,5:print space$(7)
 36: locate 60,5:print sc
 37: locate 57,7:print ball
 38: sp_slon(0):sp_disp(1)
 39: /*=== ボールの残数チェック ===*/
 40: while ball<>0
 41:    /*=== サービス ===*/
 42:    if svc_flg=1 then {
 43:       locate 57,7:print ball
 44:       by=470:dy=abs(dy)*-1
 45:       repeat
 46:          bx=sp_stat(1,0):sp_loc(1):sp_loc(0,bx+8,by)
 47:       until strig(1)<>0
 48:       calc_ball(sp_stat(0,0)+8,sp_stat(0,1)+8)
 49:       /*=== ボールとブロックのヒットチェック開始 ===*/
 50:       sp_hiton(0,1)
 51:       svc_flg=0
 52:    }
 53:    /*=== ボールの状態を取得 ===*/
 54:    if (sp_xstat(0) and 1)=0 then {
 55:       bx=sp_stat(0,0):by=sp_stat(0,1)
 56:       if by<=21 then {
 57:          dy=abs(dy):speed=128:calc_ball(bx,by)
 58:       }
 59:       if bx>=385 then {
 60:          ddx=abs(ddx)*-1:calc_ball(bx,by)
 61:       }
 62:       if bx<=140 then {
 63:          ddx=abs(ddx):calc_ball(bx,by)
 64:       }
 65:    }
 66:    /*=== ボールとブロックのヒットチェック ===*/
 67:    s=sp_hit(0)
 68:    if s<>255 then {
 69:       sp_loc(0):sp_hgrmv(1,s)
 70:       bx=sp_stat(0,0):by=sp_stat(0,1)
 71:       blx=sp_stat(s,0):bly=sp_stat(s,1)
 72:       flg=1
 73:       /* 23-10=13
 74:       if bx>=blx+13 and sgn(ddx)=-1 then {
 75:          ddx=ddx*-1
 76:       } else {
 77:       /* -16+10=-6
 78:          if bx<=blx-6 and sgn(ddx)=1 then {
 79:             ddx=ddx*-1
 80:          } else {
 81:             dy=dy*-1
 82:          }
 83:       }
 84:       sp_off(s,s+1):cnt=cnt+1:calc_ball(bx,by):sp_hiton(0,1)
 85:       i=sp_intcnt():repeat:until sp_intcnt()>i+2
 86:       /*=== スコア表示 ===*/
 87:       if s>=8  and s<=26  then sc=sc+50*flg
 88:       if s>=28 and s<=46  then sc=sc+30*flg
 89:       if s>=48 and s<=66  then sc=sc+30*flg
 90:       if s>=68 and s<=106 then sc=sc+10*flg
 91:       locate 61-len(str$(sc)),5:print sc
 92:       if cnt=50 then {
 93:          sp_loc(0,sp_stat(1,0)+8,460)
 94:          display_block()
 95:          dy=abs(dy)*-1
 96:          svc_flg=1
 97:          sp_hiton(1,0)
 98:       }
 99:    }
100:    /*=== ボールとマイパッドのヒットチェック ===*/
101:    if (sp_xstat(0) and 1)=0 then {
102:       bx=sp_stat(0,0):by=sp_stat(0,1)
103:       if by>=473 then {
104:          sp_hiton(1,0):sp_hiton(0,1)
105:          i=sp_intcnt():repeat:until sp_intcnt()>i
106:          s=sp_hit(1)
107:          if s=255 then {
108:             calc_ball(bx,by):ball=ball-1:svc_flg=1
109:             symbol(224,216,"MISS!!",1,1,2,65515,0)
110:             wait()
111:             fill(224,216,368,240,0)
112:             sp_stkon(1,1,4,0)
113:          } else {
114:             sp_loc(0):old_ddx=ddx
115:             if bx+8>=sp_stat(1,0)+16 then {
116:                if sgn(ddx)=-1 then ddx=ddx*-1
117:             } else {
118:                if sgn(ddx)=1  then ddx=ddx*-1
119:             }
120:             switch stick(1)
121:               case  4:ddx=ddx-2
122:               case  6:ddx=ddx+2
123:               default:ddx=sgn(ddx)*2
124:             endswitch
125:             if ddx>6 then ddx=6
126:             if ddx<-6 then ddx=-6
127:             if ddx=0 then ddx=sgn(old_ddx)
128:             if bx=140 or bx=385 then ddx=ddx*-1
129:             dy=dy*-1
130:             calc_ball(bx,by)
131:          }
132:       }
133:    }
134: endwhile
135: sp_sloff()
136: locate 24,14:print "REPLAY (Y/OTHER) ";
137: while inkey$(0)<>"":endwhile:a=inkey$
138: locate 27,12:print space$(18)
139: locate 24,14:print space$(18)
140: if a="Y" or a="y" then continue else break
141: endwhile
142: sp_sloff()
143: end
144: /*
145: func wait()
146:    int i,j
147:    i=sp_intcnt()
148:    repeat
149:       j=sp_intcnt()-i
150:    until j>=60
151: endfunc
152: /*
153: /* ボールの軌道計算
154: func calc_ball(bx;int,by;int)
155:    int ox,oy
156:    if sgn(dy)=-1 then {
157:       if sgn(ddx)=1 then {
158:          oy=dy/ddx*(385-bx)+by:ox=385
159:       } else {
160:          oy=dy/ddx*(140-bx)+by:ox=140
161:       }
162:       if oy<0 then {
163:          ox=ddx/dy*(21-by)+bx:oy=21
164:       }
165:    } else {
166:       if by>=473 then {
167:          ox=ddx/dy*(507-by)+bx:oy=507
168:       } else {
169:          ox=ddx/dy*(473-by)+bx:oy=473
170:       }
171:       if ox<140 then {
172:          oy=dy/ddx*(140-bx)+by:ox=140
173:       }
174:       if ox>385 then {
175:          oy=dy/ddx*(385-bx)+by:ox=385
176:       }
177:    }
178:    sp_slidep(0,ox,oy,speed)
179: endfunc
180: /*
181: /* ブロック表示
182: func display_block()
183:    int i,j
184:    sp_on(8,109)
185:    for i=0 to 4
186:       for j=0 to 9
187:          sp_loc(8+i*20+j*2,146+j*25,120+18*i)
188:          sp_loc(9+i*20+j*2,146+j*25+16,120+18*i)
189:       next
190:    next
191: /*
192: /* blockのhit range, 接触判定グループ1設定
193:    for i=0 to 49
194:       sp_hitrng(8+i*2,0,0,23,15)
195:       sp_hgadd(1,8+i*2)
196:    next
197: /*
198:    dx=2:dy=-3:ddx=dx:cnt=0:speed=96
199: endfunc
200: /*
201: func defpat()
202:    int i,j
203:    dim char ball(255)={
204:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
205:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
206:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
207:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
208:       +0,0,0,0,0,0,15,15,15,15,0,0,0,0,0,0,
209:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
210:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
211:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
212:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
213:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
214:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
215:       +0,0,0,0,0,0,15,15,15,15,0,0,0,0,0,0,
216:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
217:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
218:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
219:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }
220: /*
221:    dim char pad(255)={
222:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
223:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
224:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
225:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
226:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
227:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
228:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
229:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
230:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
231:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
232:       +15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,
233:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
234:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
235:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
236:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
237:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }
238:    dim char dum(255),blk1(255),blk2(255)
239: /*
240:    for i=0 to 255:dum(i)=0:blk2(i)=0:next
```

▶Oh!Xの値上げで定期購読にしようと思う今日この頃……。　岡本　健太郎(17)愛知県

```
241:    for i=0 to 255:blk1(i)=10:next
242:    for i=0 to 15
243:       for j=0 to 7
244:          blk2(i*16+j)=10
245:       next
246:    next
247:    sp_def(0,ball):sp_move(0,,,0)
248:    sp_def(1,pad):for i=1 to 2:sp_move(i,,,1):next
249:    sp_def(3,blk1):sp_def(4,blk2)
250:    for i=0 to 4
251:       for j=0 to 9
252:          sp_move(8+i*20+j*2,,,3):sp_move(9+i*20+j*2,,,4)
253:       next
254:    next
255: endfunc
```

リスト2

```
  1: /*
  2: /* バトルテニス
  3: /*
  4: str a
  5: char s=1 /* hit check work
  6: char svc_flg=1 /* service flag
  7: char old_s
  8: char speed=128
  9: float f
 10: float dx=2   /* ball speed  vx
 11: float dy=-3 /* ball speed  vy
 12: float bx,by
 13: int ov=160  /* other pad speed
 14: int ddx,old_dy
 15: int sx,sy=474
 16: int ex=255 /* other pad start x
 17: int sc,osc
 18: int w1,w2
 19: screen 1,3,1,1:console 0,31,0:sp_init():sp_xinit():defpat()
 20: sp_hitrng(0,3,3,12,12)
 21: sp_hitrng(1,5,0,10,31)
 22: sp_hitrng(3,5,0,10,31)
 23: sp_hang(1,2,0,16) /*    my pad connect
 24: sp_hang(3,4,0,16) /* other pad connect
 25: /*
 26: apage(0):fill(32,44,48,479,1)
 27: for i=0 to 8
 28:    box(8-i,31-i,503+i,492+i,65535)
 29: next
 30: sp_loc(1,48,272) /* display my pad
 31: sp_loc(3,480,272)  /* display other pad
 32: /*
 33: sp_stkon(1,1,0,6)
 34: sp_hgadd(0,1):sp_hgadd(0,3)
 35: sp_hiton(0,0)
 36: while 1
 37: cls:sc=0:osc=0:ddx=dx:display_score()
 38: sp_slon(0):sp_disp(1)
 39: /*=== 7点先取でおしまい ===*/
 40: while sc<>7 and osc<>7
 41:    if svc_flg=1 then {
 42:       speed=96:dy=abs(dy)*-1:sp_loc(0):sx=sp_stat(s,0)+8
 43:       repeat
 44:          sp_loc(0,sx,sp_stat(s,1)+8)
 45:       until strig(1)<>0
 46:       calc_ball(sx,sp_stat(s,1)+8)
 47:       svc_flg=0
 48:    }
 49:    /*=== ボールの状態を取得 ===*/
 50:    if (sp_xstat(0) and 1)=0 then {
 51:       bx=sp_stat(0,0):by=sp_stat(0,1)
 52:       if by=44 then {
 53:          old_dy=abs(dy)*-1:dy=abs(dy):calc_ball(bx,by)
 54:       }
 55:       if by=496 then {
 56:          old_dy=abs(dy):dy=abs(dy)*-1:calc_ball(bx,by)
 57:       }
 58:       if bx=54 then {
 59:          /*=== ボールとマイパッドのヒットチェック ===*/
 60:          if sp_hit(0)=1 then {
 61:             speed=speed+4
 62:             ddx=ddx*-1:calc_ball(54,sp_stat(0,1))
 63:          } else {
 64:             oy=dy/ddx*(38-bx)+by:ox=38
 65:             sp_slidep(0,ox,oy,speed)
 66:             osc=osc+1:ddx=abs(dx)
 67:             display_score()
 68:             svc_flg=1:wait():sp_stkon(1,1,0,6)
 69:          }
 70:       }
 71:       if bx=474 then {
 72:       /*=== ボールとアザーパッドのヒットチェック ===*/
 73:          if sp_hit(0)=3 then {
 74:             speed=speed+4
 75:             ddx=ddx*-1:calc_ball(474,sp_stat(0,1))
 76:          } else {
 77:             oy=dy/ddx*(490-bx)+by:ox=490
 78:             sp_slidep(0,ox,oy,speed)
 79:             sp_loc(3):sc=sc+1:ddx=abs(dx)*-1
 80:             display_score()
 81:             svc_flg=1:wait():sp_stkon(1,1,0,6)
 82:          }
 83:       }
 84:    }
 85:    if sp_stat(0,0)>=180 and sp_stat(0,0)<=200 then {
 86:       sp_hiton(0,0)
 87:    }
 88:    /*=== CPUのパドルの移動 ===*/
 89:    if dy<>old_dy then {
 90:       bx=sp_stat(0,0):by=sp_stat(0,1) /* ボールの座標
 91:       by=by-48:oy=dy/ddx*(480-bx)+by+40
 92:       if oy>-96 and oy<48 then oy=48
 93:       if oy>479 and oy<607 then oy=477
 94:       if oy>=32 and oy<=477 then {
 95:          w1=rnd()*5:w2=sp_intcnt()
 96:          repeat:until sp_intcnt()-w2>=w1
 97:          sp_slidep(3,480,oy,ov)
 98:       }
 99:       old_dy=dy
100:    }
101: endwhile
102: sp_sloff()
103: if sc=7 then {
104:    locate 27,12:print "YOU WIN!!"
105: } else {
106:    locate 27,12:print "YOU LOSE"
107: }
108: locate 24,14:print "REPLAY (Y/OTHER) ";
109: while inkey$(0)<>"":endwhile:a=inkey$
110: locate 27,12:print space$(18)
111: locate 24,14:print space$(18)
112: if a="Y" or a="y" then continue else break
113: endwhile
114: end
115: /*
116: func display_score()
117:    /*=== スコア表示 ===
118:    locate sc*2,0:for i=1 to sc:print "●";:next
119:    locate 62-osc*2,0:for i=1 to osc:print "●";:next
120: endfunc
121: /*
122: /* ボールの軌道計算
123: func calc_ball(bx;int,by;int)
124:    int ox,oy
125:    if sgn(dy)=-1 then {
126:       ox=ddx/dy*(44-by)+bx:oy=44
127:    } else {
128:       ox=ddx/dy*(496-by)+bx:oy=496
129:    }
130:    if ox>474 then {
131:       oy=dy/ddx*(474-bx)+by:ox=474
132:    }
133:    if ox<54 then {
134:       oy=dy/ddx*(54-bx)+by:ox=54
135:    }
136:    sp_slidep(0,ox,oy,speed)
137: endfunc
138: /*
139: func wait()
140:    int i,j
141:    i=sp_intcnt()
142:    repeat
143:       j=sp_intcnt()-i
144:    until j>=60
145: endfunc
146: /*
147: func defpat()
148:    int i,j
149:    dim char ball(255)={
150:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
151:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
152:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
153:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
154:       +0,0,0,0,0,0,15,15,15,15,0,0,0,0,0,0,
155:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
156:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
157:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
158:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
159:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
160:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
161:       +0,0,0,0,0,0,15,15,15,15,0,0,0,0,0,0,
162:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
163:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
164:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
165:       +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }
166: /*
167:    dim char pad(255)={
168:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
169:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
170:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
171:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
172:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
173:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
174:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
175:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
176:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
177:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
178:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
179:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
180:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
181:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
182:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0,
183:       +0,0,0,0,0,15,15,15,15,15,15,0,0,0,0,0 }
184: /*
185:    dim char dum(255)
186: /*
187:    sp_def(0,ball):sp_move(0,,,0)
188:    sp_def(1,pad):for i=1 to 4:sp_move(i,,,1):next
189: endfunc
```

▶初心者がX68030を使うのって無謀なんでしょうか……？　でも，こわいくらい快適。
菅井　大輔(20)宮城県

理由はないけどエイリアンをぶちのめせ！

GALAX

Kikuchi Isao 菊地 功

お次はギャラクシアンタイプのシューティングゲームGALAX
プレイヤーは迫りくるエイリアンを迎撃するのだ！
簡単そうに見えても全20ラウンドをクリアするのは結構難しいぞ

引き受けなきゃよかったかな。専門分野外の原稿依頼にちょっとだけ後悔してしまったかも。だいたい私はほとんどゲームなんか作ったことないし，定石なんかも知らないもんで，参考になるとこなんてないかもしれません。

しかし，依頼を受けてしまったからにはなにか作らなくてはなりません。はて，なにを作ろうかと遠い昔に思いを馳せたわけなんですが，どれも誌面に載せる分量で収まりそうもない&自分の技術が及ばないものばかりで，結局ギャラクシアンタイプというポピュラーなものに落ちついてしまいました。

プログラムはC言語で書かれているので，コンパイルできる環境にない方は，残念ながら遊ぶことができません。また，GCCとXCの両方でコンパイルのテストをしましたが，ライブラリはXCのものを使用しましたので，libcでコンパイルできるかどうかはチェックしていません。

まず，付録ディスクから解凍したgalax.cをコンパイルしてください。

GCCの場合は，

BASLIB,IOCSLIB,DOSLIB,FLOATFNC(FLOATEML)ライブラリをリンクします。

XCの場合は，

/O /Y /W

オプションをつけてコンパイルしてください。

ブ～ンと飛来するエイリアンを撃て！

次にスプライトデータのgalax.datを，コンパイルしてできたgalax.xと一緒にカレントに置いて実行するだけです。

遊び方

変な形の戦闘機を駆って，画面上方にたむろする蠅だか蚊だかわからんエイリアンを殲滅（せんめつ）してください。自機の左右移動はテンキーの4，6，もしくはジョイスティックの左右，ミサイル発射はスペースキーかジョイスティックのAボタンです。ちょっとすると，しびれを切らしたエイリアンが特攻をかけてきますので迎え撃ってあげましょう。

すべてのエイリアンを全滅させると1面クリアです。ボスがいて小編隊を組んで襲ってきたりはしませんが，面が進むにつれだんだん敵の攻撃が激しくなり，怒って赤くなり始めます。こいつらは1発食らうたびに冷静さを取り戻しますので，青ざめさせてから殺してあげましょう。得点は以下のようになっています。

- 青…………50点
- 青紫………100点
- 赤紫………150点
- 赤…………200点

また，特攻中に落とすと得点が倍になり，1回の特攻で2発以上当てて落とすと，それに応じたボーナスが加算されます。自機は5機で，ゲームスタート時にSHIFTキーまたは，ひらがなキーをONにしておくことで，コンティニューすることができます。

そして，20面が敵の色分けの1サイクルになっていますので，がんばって20面まではクリアしましょう。

プログラムについて

このプログラム，元はmain()関数1つしかなかったのですが，作っている途中でコンパイラのスタックが足りなくなってしまいました。そこでやむなく適当に関数に分けたのですが，おかげでgaluxy()関数が，ゲームオーバーしようがステージの開始時にしか帰ってこない変な仕様になってしまいました。計画性もなしに勢いだけでプログラムすると，こうなってしまいます。みなさんも気をつけましょう（説得力がないけど）。

今回は敵の編隊をバックグラウンドで描いてみました。隊列を乱さずに一斉に移動するのだから，このほうがいいかな，と思ったのですが，特攻するやつは結局スプライトでやらなければならないので，あまり意味がなかったかもしれません。その関係で，敵が羽をばたつかせるのは，PCGエリアを直接書き換えることでアニメーション（？）させています。

バックグラウンドを全部書き換えるよりこのほうがいいや，と思ったのですが，いま考えると「そうか？」と疑問に感じてしまいます。バックグラウンドはもう1枚余っているので，あらかじめそちらにも別パターンを定義しておいて，パタパタと切り替えたほうがよかったんでしょうねぇ，きっと。あとの祭りですが。

敵が特攻してきたり，弾を撃ってきたりするのは，面数や倒した敵の数などの値から，ランダムに決定しています。しかし，rnd()をメインループ内から呼ぶと，遅くなりそうだったので，起動するとまず配列にrnd()で得られたデータを読み込んでいます。そのため，起動に若干時間がかかりますので，飛んだと思ってインタラプトを押さないでくださいね。

敵や敵の弾の座標ですが，6ビットの下駄を履かせて演算しています。こうしないと，精度が落ちてしまい，弾がちゃんと自機へ向かってこなかったり，敵の特攻の軌跡が折れ線状になってしまい，「だっせ～」ってことになってしまうのです。

また，1回のループでそう何回も呼ばれることはないので，敵の弾の移動量を求めるのにsqrt()なんて関数を使っていますが，こういった浮動小数点を扱う関数は多用すべきではありません。

では，関数ごとに簡単に説明をしていきましょう。

●main()関数

画面の初期化，ランダムデータの作成，スプライトデータの読み込みと設定，およびステージごとの変数初期化とgaluxy()関数の呼び出しを行っています。スプライトデータは，ファイルから呼んでPCGに流し込んでいます。ステージのループの中で，最初にgaluxy()が呼ばれているのは，タイトル画面（？）やゲームオーバー画面で背景をスクロールさせるのもgaluxy()にやらせているからです。これもgaluxy()関数の変な仕様のせいですね。敵編隊初期化では，面数に応じて硬さを設定しています。

●galuxy()関数

メインループです。ここでは変数statusによって挙動が変わります。

・status＝0

通常の状態です。自機も自由に動くことができれば，敵も攻撃を仕掛けてくることができます。敵を落とした数hitcountが24に達すると，status＝1に移行します。

・status＝1

敵を全滅させて，ほっとしたひとときです。ステージナンバーや“Ready”という文字を表示させながら，ループを300回まわして親関数に戻っていきます（本当に変な関数だなぁ）。

・status＝2

自機が煙を上げて爆発しているところです。ループを40回ほどまわしたあとに，status＝4に移行します。また，このときはすでに飛び立った敵は動けますが，新たに攻撃を仕掛けてくることはしません。

・status＝3

自機が煙とともに消失し，がっくりきているところです。“Ready”という文字を表示させながら，ループを200回まわし，残機がある場合にはstatus＝0に移行しゲーム続行，ない場合にはstatus＝4に移行しゲームオーバーです。status＝2同様敵は攻撃を仕掛けてきません。

・status＝4

スペースキーかジョイスティックのAボタンが押されて，ゲームが開始されるのを待ちます。このとき画面に残っている敵は，ただひたすら反復横飛びを続けます。

それぞれの状態で，演算しなければならない処理を行ったあと，垂直帰線期間を待って，それぞれのスプライトやバックグラウンドを移動させます。このあたりは定石ですね。スプライトコード8の敵のパターンの書き換えや，背景のスクロールもここで行っています。

●myship()関数

自機の操作と，発射されていればミサイルの移動を行っています。

●group()関数

だんだん関数名の意味がなくなってきましたが，敵の群れの移動と，ミサイルとの当たり判定をしています。敵の左右の移動は，まだエイリアンが現存するエリアが画面端にぶつかると，移動方向が反転するようになっています。ミサイルとの当たり判定は，群れでまとめて判定していますが，エイリアン同士は密接しておらず，隙間が空いているので，その間をミサイルがすり抜けていないかも判定しています。

・attack()関数

エイリアンを飛び立たせる関数です。乱数で適当に出撃させるかどうか決定し，出撃するのであれば，何列目の左右どちら側からかをさらに乱数で決定します。つまり真ん中から突然飛び出すことはないわけですね。そのあと，自機との距離から適当に初期移動量，その他もろもろを決定します。変数mについてですが，これは乱数の偏り（？）でしばらく誰も飛び出さなかった場合，あるいは決定された列に誰もいなかった場合に，次のループですぐに代理を出すための調整です（いいかげん）。

●flingenemy()関数

飛んでるエイリアン関連の関数です。まずはエイリアンを移動させ，Y座標が96になったら乱数で弾を発射させます。つまり弾はいつも同じ高さで発射されるんですね。あとは，エイリアンと自機の当たり判定，エイリアンとミサイルの当たり判定，エイリアンの弾の移動と，最後に弾と自機の当たり判定です。

●combo()関数

飛来中のエイリアンを撃墜したときの得点と，爆発なり得点なりの表示の時間設定です。フラグをカウンタ代わりに使うなんて，極悪非道かも（すでにひとごと）。

●bgenemy()関数

バックグラウンドにスプライトパターンを張りつける関数です。画面モードが256×256なんで，バックグラウンドのパターンサイズは8×8なんですよね。エイリアンのパターンサイズは16×16なので，4枚まとめて貼れるように関数化しました。

遊ぶための注意

XCでもコンパイルできると書きましたが，情けないことに10MHz機では処理が間に合わず，重くなってしまいます。できればGCCでコンパイルしてほしいのですが，どうしてもXCで，という方はリストの213行を削除して，垂直帰線期間を待たないようにしてください。こうしておけば，多少ちらついたり，速度にばらつきがあったりするものの，比較的まともにプレイできます。

また，254行のコメントですが，1回の垂直帰線期間でループを1回転以上してしまうという鬼のように速いマシンでは（あるのかそんなの），先頭の“/＊”を削除してください。

垂直帰線期間を待たないようにして，XCとGCCでコンパイルしたものを比べてみて，どれほど違うか体感してみるのもいいかもしれません。

そうそう，このゲームを実行するときにはHIOCSを組み込んでおいてくださいね。printf()でエスケープシーケンスを使って位置出しとかしてますので。

では，懐かしい雰囲気を噛み締めつつ，楽しんでください。

リスト

```
  1: #include      <stdlib.h>
  2: #include      <stdio.h>
  3: #include      <sprite.h>
  4: #include      <doslib.h>
  5: #include      <iocslib.h>
  6: #include      <time.h>
  7: #include      <basic.h>
  8: #include      <graph.h>
  9: #include      <math.h>
 10:
 11: #define PARM1   1000 /* 小さくすると敵が飛んでくる確率が高くなる */
 12: #define PARM2   50   /* 小さくすると敵が弾を出す確率が高くなる */
 13:
 14: int     galuxy();
 15:
 16: void    myship();
 17: void    group( int );
 18: int     attack( int );
 19: void    flingenemy();
 20: void    combo( int, int );
 21: void    bgenemy( int, int, int );
 22:
 23: int     x0;             /* 自機のX座標 */
 24: int     x1, y1;         /* 自分のミサイルの座標 */
 25: int     mflag;          /* ミサイル不在で立つフラグ */
 26: int     x2;             /* 敵編隊のX座標 */
 27: int     vx;             /* 敵編隊の移動方向 */
 28: int     hitcount;       /* ヒットカウント */
 29: int     score;          /* スコア */
 30: int     by0, by1;       /* 背景のy座標 */
 31: int     status;         /* 状態 0:通常 1:ステージクリア */
 32:                         /* 2:自分の爆発中 3:自分がやられた後 */
 33: int     count;          /* 状態変移のカウンタ */
 34: int     stage;          /* ステージナンバー */
 35: int     rest;           /* 残機数 */
 36: short   rndtbl[65536];  /* ランダムテーブル */
 37: unsigned short  rndcnt; /* ランダムカウンタ */
 38:
 39: struct {                /* 敵の情報 */
 40:         char    flag;   /* 存在情報 1:御在宅 -1:出張中 */
 41:                         /* 0:おまえは既に死んでいる */
 42:                         /* -2:帰還中 2:爆発中 */
 43:         int     x, y;   /* 出張先 */
 44:         int     x0;     /* 64倍仮想X座標 */
 45:         int     vx, vy; /* 移動量 */
 46:         int     vvx;    /* X移動量の増加量 */
 47:         char    10, 1;  /* 最初のかたさと今のかたさ */
 48:         char    11;     /* 出張前のかたさ */
 49:         char    bflag;  /* 弾を吐いたフラグ */
 50:         int     bx, by; /* 弾の座標 */
 51:         int     bx0, by0;       /* 64倍座標 */
 52:         int     vbx, vby;       /* 弾の移動量 */
 53: } enemy[8][3];
 54:
 55: unsigned short  enemypat[2][64]; /* 敵のアニメーションパターン */
 56:
 57: void    main()
 58: {
 59:   int  x, y;
 60:   unsigned short  *sp;
 61:   FILE  *fp;
 62:   int  ssp, i;
 63:   unsigned short  dat;
 64:
 65:   for( i=0; i<65536; i++ ) rndtbl[i] = rnd()*PARM1;
 66:   CRTMOD( 6 );
 67:   G_CLR_ON();
 68:   OS_CUROF();
 69:   window( 0, 0, 511, 511 );
 70:   randomize( time( NULL ) );
 71:   vpage( 3 );
 72:   apage( 0 );
 73:   for( i=0; i<100; i++ ) pset( rnd()*256, rnd()*512, rnd()*16 );
 74:   apage( 1 );
 75:   for( i=0; i<100; i++ ) pset( rnd()*256, rnd()*512, rnd()*16 );
 76:
 77:   sp_init();
 78:   ssp = SUPER( 0 );
 79:   fp = fopen( "galuxy.dat", "rb" );
 80:   sp = (unsigned short *)0xEB8080;
 81:   for( i=0; i<64*7; i++ ){
 82:     dat = fgetc( fp )<<8;
 83:     dat |= fgetc( fp );
 84:     *(sp++) = dat;
 85:   }
 86:   for( i=0; i<64*2; i++ ){
 87:     dat = fgetc( fp )<<8;
 88:     dat |= fgetc( fp );
 89:     *(sp++) = dat;
 90:     enemypat[i/64][i%64] = dat;
 91:   }
 92:   sp = (unsigned short *)0xE82220;
 93:   for( i=0; i<16*4; i++ ){
 94:     dat = fgetc( fp )<<8;
 95:     dat |= fgetc( fp );
 96:     *(sp++) = dat;
 97:   }
 98:
 99:   fclose( fp );
100:
101:   by0 = by1 = 512;
102:   sp_disp( 1 );
103:   printf( "¥x1B[1;1HSCORE" );
104:   printf( "¥x1B[1;24HSTAGE" );
105:   printf( "¥x1B[8;11H≪Galuxy≫" );
106:   stage = 1;
107:   status = 4;
108:   for( ;; ){
109:     if( galuxy() ) break;
110:     bg_set( 0, 0, 0 );
111:     for( y=0; y<3; y++ ) for( x=0; x<8; x++ ){  /* 敵編隊初期化 */
112:      enemy[x][y].flag = 1;
113:      switch( y ){
114:      case 0:  if( stage%20<=2 ) i = 1; else if( stage%20<=6 ) i
= 2;
115:        else if( stage%20<=12 ) i = 3; else i = 4;
116:        break;
117:      case 1:  if( stage%20<=4 ) i = 1; else if( stage%20<=10 ) i
= 2;
118:        else if( stage%20<=16 ) i = 3; else i = 4;
119:       break;
120:      case 2:  if( stage%20<=8 ) i = 1; else if( stage%20<=14 ) i
= 2;
121:       else if( stage%20<=18 ) i = 3; else i = 4;
122:      }
123:       enemy[x][y].10 = enemy[x][y].1 = i;
124:       enemy[x][y].bflag = 0;
125:       bgenemy( x, y, 8 );
126:     }
127:     x2 = 36;
128:     bg_scroll( 0, 512-x2, 512-16 );
129:     bg_set( 0, 0, 1 );
130:     vx = 1;
131:     status = 0;
132:   }
133:   SUPER( ssp );
134:   CRTMOD( 16 );
135:   OS_CURON();
136:   KFLUSHIO( 0xFF );  /* キーバッファをフラッシュする */
137: }
138:
139: int  galuxy()
140: {
141:   int  i, j, k, am, m, lr, sno;
142:   volatile char *const mfp = 0xe88001;
143:   unsigned short  *sp;
144:
145:   hitcount = 0;
146:   am = 0;
147:   m = 0;
148:   lr =0;
149:   for( i=0;; i++ ){
150:     switch( status ){
151:     case 0:  /* 通常状態 */
152:       if( hitcount>=24 ){
153:         status = 1;
154:         count = 0;
155:       }
156:       break;
157:     case 1: /* クリアした後 */
158:       if( count==100 ){
159:         printf( "¥x1B[8;12HSTAGE %3d", ++stage );
160:         printf( "¥x1B[1;30H%03d", stage );
161:       }
162:       if( count==200 ) printf( "¥x1B[8;12H  Ready  " );
163:       if( ++count>=300 ){
164:         printf( "¥x1B[8;14H¥x1B[0K" );
165:         return( 0 );
166:       }
167:       break;
168:     case 2:  /* 自機の爆発の最中 */
169:       if( ++count>=40 ){
170:         status = 3;
171:         count = 0;
172:       }
173:       break;
174:     case 3:  /* 自機が爆発した後 */
175:       if( rest && count==100 ) printf( "¥x1B[8;14HReady" );
176:       if( ++count>=200 ){
177:         if( rest ){
178:           rest--;
179:           status = 0;
180:           m = 0;
181:           x0 = 136;
182:           x1 = 136; y1 = 236;
183:           printf( "¥x1B[16;1H¥x1B[0K" );
184:           for( j=0; j<rest; j++ ) printf( "△" );
185:           printf( "¥x1B[8;14H¥x1B[0K" );
186:         } else {
187:           status = 4;
188:           printf( "¥x1B[8;12HGAME OVER" );
189:         }
190:       }
191:       break;
192:     case 4:  /* ゲーム開始待ち */
193:       if( !(JOYGET( 0 )&0x20) || K_KEYBIT( 6 )&0x20 ){
194:         printf( "¥x1B[1;7H000000" );
195:         printf( "¥x1B[8;11H¥x1B[0K¥x1B[16;1H△△△△" );
196:         if( K_SFTSNS()&0x201 ) stage--; else stage = 0;
197:         bg_set( 0, 0, 0 );
198:         rest = 4;
199:         score = 0;
200:         count = 100;
201:         status = 1;
202:         x0 = 136;
203:         x1 = 136; y1 = 236;
204:       }
205:       if( K_KEYBIT( 0 )&2 ) return( 1 );  /* ESC */
206:     }
207:
208:     myship();
209:     if( status!=1 ) group( i );
210:     if( status==0 ) m = attack( m );
211:     flingenemy();
212:
213:     while( (*mfp)&0x10 );  /* 垂直帰線期間を待つ */
214:     if( i%8==0 ){  /* アニメーションというほどでもないが */
215:       am ^= 1;
216:       sp = (unsigned short *)0xEB8400;  /* SPコード8 敵 */
217:       for( j=0; j<64; j++ ) *(sp++) = enemypat[am][j];
218:     }
219:     bg_scroll( 0, 512-x2, 512-16 );
220:                                                /* 自機の表示 */
221:     if( status==2 ) sp_set( 0, x0, 240, 256+4, 3 );
222:     else if( status>=3 ) sp_set( 0, x0, 240, 0, 3 );
223:     else sp_set( 0, x0, 240, 256+1, 3 );
224:                                                /* ミサイルの表示 */
225:     if( status<=1 || mflag ) sp_set( 1, x1, y1, 256+2, 3 );
226:     else sp_set( 1, x1, y1, 0, 3 );
227:     sno = 1;
228:     for( j=0; j<8; j++ ) for( k=0; k<3; k++ ){
229:       if( enemy[j][k].flag<0 ){     /* 出張中の敵の表示 */
230:         if( enemy[j][k].x>=0 && enemy[j][k].x<272 ){
231:           sno++;
```

▶友人(12Mバイトユーザー)がこのはがきを毎月出してメモリの平均を上げるといっていたので，私は平均メモリを下げたいと思います(2Mバイト)。　新原　恵太(20)埼玉県

```
232:           sp_set( sno, enemy[j][k].x, enemy[j][k].y,
233:             (enemy[j][k].l<<8)|0x8008, 3 );
234:         }
235:       }
236:       if( enemy[j][k].flag>=2 ){     /* 爆発中の敵の表示 */
237:         sno++;
238:         sp_set( sno, enemy[j][k].x, enemy[j][k].y,
239:           256+enemy[j][k].ll+3, 3 );
240:         if( --enemy[j][k].flag<2 ){
241:           enemy[j][k].flag = 0;
242:           hitcount++;
243:         }
244:       }
245:       if( enemy[j][k].bflag )        /* 敵の弾の表示 */
246:       sp_set( ++sno, enemy[j][k].bx, enemy[j][k].by, 256+3, 3 );
247:     }
248:     sp_on( 0, sno );
249:     sp_off( sno+1, 127 );
250:     HOME( 0, 0, by0-=2 );
251:     HOME( 1, 0, by1-=1 );
252:     if( by0==0 ) by0 = 512;
253:     if( by1==0 ) by1 = 512;
254: /*     while( !((*mfp)&0x10) );  /* 垂直表示期間を待つ */
255:   }
256: }
257:
258: void  myship()
259: /* 自機の操作 */
260: {
261:   int  i;
262:
263:   if( status<=1 ){
264:     i = JOYGET( 0 );
265:     /* 右 */
266:     if( !(i&8) || K_KEYBIT( 9 )&2 ) if( x0<255 ) x0 += 1;
267:     /* 左 */
268:     if( !(i&4) || K_KEYBIT( 8 )&0x80 ) if( x0>16 ) x0 -= 1;
269:     if( !mflag ){
270:       /* ボタン */
271:       if( !(i&0x20) || K_KEYBIT( 6 )&0x20 ) mflag = 1;
272:       else x1 = x0;
273:     }
274:   }
275:   if( mflag ){
276:     y1 -= 6;
277:     if( y1<=0 ){
278:       mflag = 0;
279:       x1 = x0;
280:       y1 = 236;
281:     }
282:   }
283: }
284:
285: void  group( int i )
286: /* 敵の群れの移動およびミサイルとの当たり判定 */
287: {
288:   int  j, k;
289:
290:   if( i%4==0 ){  /* 4週りに1回 */
291:     if( vx>0 ){  /* 右移動 */
292:       for( j=7; j>=0; j-- ) if( enemy[j][0].flag!=0 ||
293:       enemy[j][1].flag!=0 || enemy[j][2].flag!=0 ) break;
294:       if( x2>=256-16-24*j ) vx = -1;
295:     } else {  /* 左移動 */
296:       for( j=0; j<8; j++ ) if( enemy[j][0].flag!=0 ||
297:       enemy[j][1].flag!=0 || enemy[j][2].flag!=0 ) break;
298:       if( x2+24*j<=0 ) vx = 1;
299:     }
300:     x2 += vx;
301:   }
302:   if( x1-8>=x2 && x1-8<x2+184 && y1-8>=16 && y1-8<16+64 ){
303:     j = (x1-8-x2)%24;
304:     k = (y1-8-16)%24;
305:     if( j<16 && k<16 ){
306:              /* キャラクタの間をすり抜けていないかチェック */
307:       j = (x1-8-x2)/24;
308:       k = (y1-8-16)/24;
309:       if( enemy[j][k].flag==1 ){  /* Hit */
310:         mflag = 0;
311:         x1 = x0;
312:         y1 = 236;
313:         if( (--enemy[j][k].l)==0 ){
314:           bgenemy( j, k, 0 );
315:           enemy[j][k].flag = 10;
316:           enemy[j][k].x = x2+j*24+16;
317:           enemy[j][k].y = 32+k*24;
318:           enemy[j][k].ll = 1;
319:           score += enemy[j][k].l0*50;
320:           printf( "¥x1B[1;7H%06d", score );
321:         } else bgenemy( j, k, 8 );
322:       }
323:     }
324:   }
325: }
326:
327: int  attack( int m )
328: /* 敵の出撃 */
329: {
330:   int  j, k;
331:
332:   m++;
333:   if( rndtbl[rndcnt++]<stage+(hitcount>>2)+m/50 ){
334:                                          /* だれか出張させる */
335:     k = rndtbl[rndcnt++]%3;
336:     if( rndtbl[rndcnt++]&1 ){  /* 左から */
337:       for( j=0; j<8; j++ ) if( enemy[j][k].flag==1 ) break;
338:       if( j>=8 ) return( 50*PARM1 );
339:       enemy[j][k].vx = -64;
340:     } else {  /* 右から */
341:       for( j=7; j>=0; j-- ) if( enemy[j][k].flag==1 ) break;
342:       if( j<0 ) return( 50*PARM1 );
343:       enemy[j][k].vx = 64;
344:     }
345:     m = 0;
346:     enemy[j][k].vy = -6;
347:     enemy[j][k].flag = -1;
348:     enemy[j][k].x0 = ((x2+j*24+16)<<6)+32;
349:     enemy[j][k].x = x2+j*24+16;
350:     enemy[j][k].y = 32+k*24;
351:     enemy[j][k].ll = enemy[j][k].l;
352:     if( enemy[j][k].x<x0 ){
353:       if( x0-enemy[j][k].x>64 ) enemy[j][k].vvx = 2;
354:       else enemy[j][k].vvx = 1;
355:     } else {
356:       if( enemy[j][k].x-x0>64 ) enemy[j][k].vvx = -2;
357:       else enemy[j][k].vvx = -1;
358:     }
359:     bgenemy( j, k, 0 );
360:   }
361:   return( m );
362: }
363:
364: void  flingenemy()
365: /* 出撃中の敵とその弾の挙動および自機との当たり判定 */
366: {
367:   int  i, j, k;
368:
369:   for( j=0; j<8; j++ ) for( k=0; k<3; k++ ){
370:     if( enemy[j][k].flag==-1 ){  /* 出張中の挙動 */
371:       enemy[j][k].x0 += enemy[j][k].vx;
372:       enemy[j][k].x = enemy[j][k].x0>>6;
373:       enemy[j][k].y += enemy[j][k].vy;
374:       enemy[j][k].vx += enemy[j][k].vvx;
375:       if( enemy[j][k].vy<2 ) enemy[j][k].vy++;
376:       if( enemy[j][k].y>272 ){
377:         enemy[j][k].flag = -2;
378:         enemy[j][k].x0 = ((x2+j*24+16)<<6)+32;
379:         enemy[j][k].x = x2+j*24+16;
380:         enemy[j][k].y = 0;
381:       }
382:       if( status==0 && enemy[j][k].y==96 && enemy[j][k].bflag==0
 ){
383:         if( (rndtbl[rndcnt++]%PARM2)<stage+(hitcount>>2) ){
384:           enemy[j][k].bflag = 1;
385:           enemy[j][k].bx0 = enemy[j][k].x<<6;
386:           enemy[j][k].by0 = 96<<6;
387:           i = x0-enemy[j][k].x;
388:           i = (int)sqrt((240-96)*(240-96)+i*i);
389:           enemy[j][k].vbx = (x0-enemy[j][k].x)*192/i;
390:           enemy[j][k].vby = (240-96)*192/i;
391:         }
392:       }
393:       if( status==0 && x0>=enemy[j][k].x-12 && x0<enemy[j][k].x+
12 &&
394:       240>=enemy[j][k].y-12 && 240<enemy[j][k].y+12 ){
395:         status = 2;
396:         count = 0;
397:         combo( j, k );
398:         score += enemy[j][k].l0*100;
399:         printf( "¥x1B[1;7H%06d", score );
400:       }
401:     } else if( enemy[j][k].flag==-2 ){
402:       enemy[j][k].x0 = ((x2+j*24+16)<<6)+32;
403:       enemy[j][k].x = x2+j*24+16;
404:       enemy[j][k].y += 1;
405:       if( enemy[j][k].y==32+k*24 ){
406:         enemy[j][k].flag = 1;
407:         bgenemy( j, k, 8 );
408:       }
409:     }
410:     if( enemy[j][k].flag<0 && mflag ){
411:       if( x1>=enemy[j][k].x-8 && x1<enemy[j][k].x+8 &&
412:       y1>=enemy[j][k].y-8 && y1<enemy[j][k].y+8 ){
413:         mflag = 0;
414:         x1 = x0;
415:         y1 = 236;
416:         if( --enemy[j][k].l==0 ){
417:           combo( j, k );
418:           score += enemy[j][k].l0*100;
419:           printf( "¥x1B[1;7H%06d", score );
420:         }
421:       }
422:     }
423:     if( enemy[j][k].bflag ){  /* 敵の弾 */
424:       enemy[j][k].bx0 += enemy[j][k].vbx;
425:       enemy[j][k].by0 += enemy[j][k].vby;
426:       enemy[j][k].bx = enemy[j][k].bx0>>6;
427:       enemy[j][k].by = enemy[j][k].by0>>6;
428:       if( enemy[j][k].bx<0 || enemy[j][k].bx>=272 ||
429:         enemy[j][k].by>=272 ) enemy[j][k].bflag = 0;
430:       if( status<=1 && x0>=enemy[j][k].bx-8 && x0<enemy[j][k].bx
+8 &&
431:       240>=enemy[j][k].by-8 && 240<enemy[j][k].by+8 ){
432:         status = 2;
433:         count = 0;
434:         enemy[j][k].bflag = 0;
435:       }
436:     }
437:   }
438: }
439:
440: void  combo( int x, int y )
441: /* コンボ?による加算スコアと得点表示時間の設定 */
442: {
443:   switch( enemy[x][y].ll ){
444:   case 2:  score += 100;
445:     enemy[x][y].flag = 60;
446:     break;
447:   case 3:  score += 300;
448:     enemy[x][y].flag = 60;
449:     break;
450:   case 4:  score += 500;
451:     enemy[x][y].flag = 60;
452:     break;
453:   default:enemy[x][y].flag = 10;
454:     break;
455:   }
456: }
457:
458: void  bgenemy( int x, int y, int n )
459: /* 群れ中の敵の表示 */
460: {
461:   bg_put( 0, x*3  , y*3  , (enemy[x][y].l<<8)|((n<<2)  ) );
462:   bg_put( 0, x*3  , y*3+1, (enemy[x][y].l<<8)|((n<<2)+1) );
463:   bg_put( 0, x*3+1, y*3  , (enemy[x][y].l<<8)|((n<<2)+2) );
464:   bg_put( 0, x*3+1, y*3+1, (enemy[x][y].l<<8)|((n<<2)+3) );
465: }
```

▶最近「1日1度はリッジレーサー」が日課になっている。中級はどうにかトップで走れるようになったのだが，上級となると，まともに完走することすらできない。T.Tで走れる日はくるのだろうか。うーん……。

城間　裕樹(20)大阪府

元気なオタッキー作法講座(完結編?)

逆襲のバニーさん

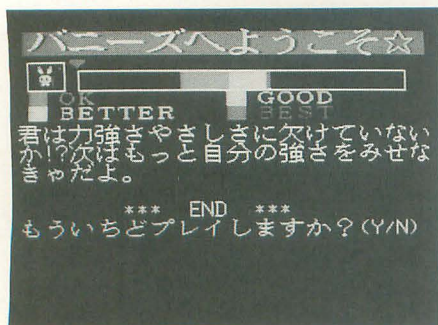
Komura Satoshi 古村 聡

**古村氏が贈る，自らの体験をもとにしたバニーシミュレータ(?)
君は，バニーさんをお持ち帰りするという夢を達成しなくてはならない
大人になったと豪語する古村氏の成長ぶりをとくと拝見しよう**

どもーっ，(て)でーっす。復活して早々に飲んでまーすっ。えへへ。こないだ，ついに取っちゃったんですよぉ，バニーさんのいるお店の会員証。今日も行ってきちゃったんだもーんっ。

考えてみれば，あれはOh!MZがOh!Xに変わって1年くらいの頃。X1用のマージャンゲーム「まじゃべんちゃー・ねぎまーじゃん」を見て「バニーさんかあいいー☆」と九段南井関ビルの4Fで叫びまくった5年前。その後は，本物のバニーさんのいる店に一度連れていってもらってからというもの，金がないんで会員証を持ってる人を探しては強引に誘ってたんですよね。会員証を持ってた人はさぞ迷惑な奴だと思ったことでしょう。

そう考えると，長い年月でありました。編集室は九段から泉岳寺，さらに現在の浜町に移り，私も学校を卒業していまやりっぱな社会人。ふふふふ，これさえあればいつバニーさんに会いにいってもおっけい。昔の私とは違うのさっ(そ，そうか?)。

そうそう，バニーさんといえば，前にバニーさんネタでプログラムを作ろうとしたこともあったんですよね。そう，あれは1989年の3月号でのこと。「オムライスが食べたい」とか「リリアがかあいい☆」とか(考えてみるととんでもねーこといってた気がする)書いていた私がどういうわけか，(N)氏に「特集でプログラムの話を書きなさい」といわれて，じゃあ，ここで一発ゲームでも作ってみようか，ってことになって「元気なオタッキー作法講座(予告編)」とかいうのを書いたんですよね。

よーっし，今回は「元気なオタッキー作法講座(完結編?)」ってことで，あのとき作り損ねたバニーさんゲームを作ることにしよう。5年ぶりにぱーっといこうか!

君はバニーさんをお持ち帰りできるか?

バニーさんのすべて

ここで，実際のバニーさんがいる店というのはどういうものか解説しておきましょう。まだそういう店にいったことがない人も多いでしょうから。

「バニーさんのいる店の会員証を取ったんだよーん」っというと，みなさん，「それってカジノかなんかですか?」というのだけど，そうではないんですよね。私がいくバニーさんのいる店というのは，単なる飲み屋さんだったりするんです。

この店では，会員となってボトルをキープいたします。お店に入ると，バニーさんがついて，このキープしたお酒で水割りを作ってくれます。で，楽しく酒を飲みながらバニーさんとお話しすると。ただそれだけなんですよね。タバコ吸うならバニーさんが胸の谷間からライター出して火をつけてくれたりもしますけど，基本的に普通の飲み屋と変わんなかったりするんです。

ただ，会員制なので，若者向けの居酒屋なんかに比べると，日と時間さえ選べば客の数も少なくて静かだし，バニーさんもわりと話好きな娘が多いみたいで，話がはずんでとっても楽しいのですよ。だもんで，集団でいくときには，自分がバニーさんと楽しくお話しするために，バニーさんが水割り作りそうな場所の隣に座ってバニーさんがくるのを虎視眈眈と待ってたりするんですけど。でもって，仲間うちで話を軽く盛り上げて，それでいて，話の内容をバニーさんが加わりやすそうな話題にしておいて，バニーさんがきたらば，ニコニコとお話を始めるわけです。平和でしょう。もう大人ですからね。

ただ，この汗と涙と根性の努力も，ちょっとでも仲間だけで盛り上がったり，からみそうなヤツがいるとバニーさんはすうっと，ほかのテーブルへいってしまったりなんかしてしまうのです。

ということで，今回のゲームのコンセプトは「バニーさんと楽しくお話ししよう!」ということにいたします。名づけて「バニーさんのいる店シミュレーション」ですかね。ただ……そうですね。ゲームってのは目的があると盛り上がるものです。

しかし，バニーさんのいる店にお酒を飲みに行くのは，別にライターの入っている谷間や1カ所ほつれた網タイツを観察しているのが楽しいのではなくて(多少はあるかもしれないけど)，ただひたすらにバニーさんとお話しするために行っているのであります。特に大きな目的や利益があるわけではないのです。

しかし，RPGが最後にドラゴンを倒すように，推理物アドベンチャーでは最後に殺人犯を見つけるように，ゲームというのは最後に最大の目標があってこそ盛り上がるのです。困ったな……。

バニーさんとお話しすると盛り上がる……えーい，こうなったら，嘘だけど「話が思いっ切り盛り上がったら，バニーさんをお持ち帰りできる」ことにしてしまおう! どうだ，がぜん盛り上がってきたでしょう。

教訓1:これはゲームだ。多少の嘘はしかたない

バニーさんのエッセンス

さーてと。予告編でも話したとおり(予告編を憶えてる人っていらっしゃるんでしょーか?)，まずはグラフィックから作り，プ

ログラム作りのための気分を盛り上げるのであります。

今回必要なものは，まず，バニーさんの絵ですかね。動かすかもしれないのでスプライトで作っておきましょう。ぺたぺた。ほら，やっぱりこーゆーテーマだとグラフィックがあれば，こう，期待に胸膨らむってのがあるよね。わくわく。

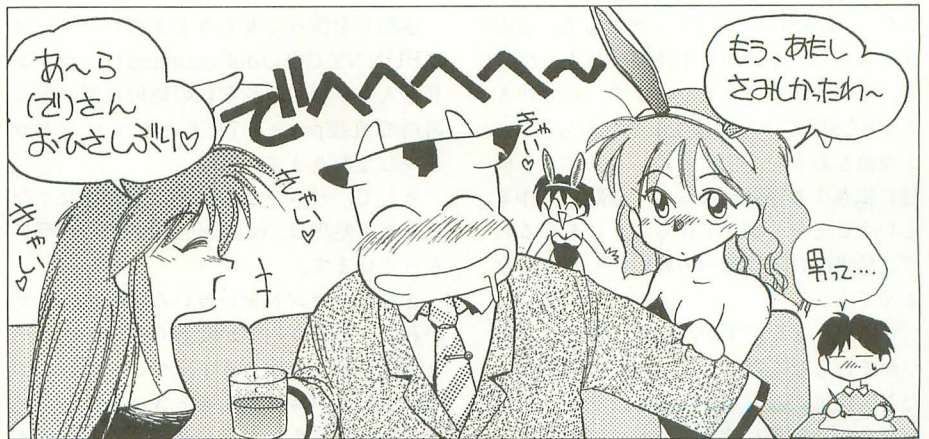
で，いよいよプログラミングのための方針を決めます。前回はBASICで行ったわけですが，今回はCで作りたいと思います。最近BASICよりCを自分がよく使ってるから，ということもあるんですが，ほら，やっぱり，たとえば，ゲームがうまくできてヒットしちゃったりなんかした場合にはCのほうが移植しやすいでありましょう。いまや，セーラームーンは世界中で放送されているそうだし，ストIIメキシコ遠征とかあったりするし，オタッキーたるもの，やはり世界を目指さなくちゃね。目標は大きくもたなくちゃいけません。

教訓2：目指せ，バニーさんで世界征服！

相手がバニーさんでは，征服されたほうは悔しくて夜も眠れないかもしれんけどね。てことでファイル名はバニーさんのプログラムだからBUNNY.Cに決まり。ヘッダファイルはBUNNY.Hだ。まあ，えっち。

さて，それでは，ひととおりグラフィックが出来上がったところで，今度は必要なパラメータとコマンド体系を設計していきましょう。特に今回のプログラムがシミュレータ(それほど厳密なものじゃないけど)ということで，どんなデータがあって，どんな型の変数，構造体やポインタを作っておくのか，というのは非常に大事なことなんであります。そして，バニーさんと話していくうえでどんなことが影響してくるか考えてみましょう。

うーんと，話をしていたバニーさんがいなくなっちゃうときって，ほかのお客さんがきたときとかもあるけど，やっぱり話題がしらけちゃったときなんだよね。それじゃ，バニーさんの興味のありそうなパラーメータを作って，そいつがあんまり低くなったらバニーさんがどっかへいっちゃう，帰るときに話が盛り上がっていたら，つまりこれらのパラメータが最高に上がっていたらバニーさんのお持ち帰りOKということにしてしまおう。

んーと，バニーさんの興味っていうと…客と…話をしていてわかること……やさしさ……力強さ……面白さ……えっちさ……えーい，これでいいや。決まり。

あとはコマンド。コマンドは，ユーザーが選んで実行すると，いま作ったパラメータに影響するようにするわけですよね。パラメータ自体がバニーさんと話をしていてバニーさんに伝わったものなんだから，プレイヤーが話す内容をメニューにしてコマンド選択するようにいたしましょう。……ほめる，笑わせる，海に行こうよとか今度映画を見に行こうよと誘う。

それから自分の話。本当は自分から自分の話なんかすると自慢たらしく聞こえるような気もするんだけど，バニーさんも「今日はどんな集まりなんですか？」とか聞いてくるから，ま，よしとしましょ。そうそう，バニーさんに水割りを作ってもらう，「水割り」コマンドも必要だな。それから，店から出る「帰る」コマンドも。

はて，まだなにかが足りない気がする。そう，お酒だお酒。バニーさんの店にいって酒を飲みながら話をするわけだけど，このお酒というのが結構話に影響するんですよ。店にいくと，バニーさんがほいほいお酒を作ってくれるわけだけど，こっちも調子にのって酒をカパカパ飲む。だもんだから，まわるんだこれが。

で，まわってまわってまわって飛んで，あらぬことを口走ったりしちゃったりするんですね。誰とはいわないんですけど，同じ店に連れてこられた人で，実際にいたんだわ。べろんべろんになるまで飲んで，バニーさんに狼藉のかぎりを尽くし「キャーっ，なにするんですかぁっ！」ってどなられて出入り禁止になったやつ。ま，私だって，酔っ払ってベラベラ変なこと話して，あとで自己嫌悪に陥ることはあるけど。私の場合は酒の上の不埒ってことで。

教訓3：他人の失敗は蜜の味。でも，刺されないように気をつけよう

これで，コマンドもできたんだけど，まだちょっと不満が残るな。だって，バニーさんと話すときって，しらふで普通に話しても，なかなかうまく話せないこともあるじゃないですか。ほら，やっぱり，自分の横に女の子がいるんだからかっこヨク話したいじゃん(なにがジャンだよ)。

でも私，シャイだからさ，なかなか思うように話せないんだよね(誰だ「ダウト」とかゆーてるやつは)。だから，しらふでも失敗すると不埒な行動とって，酔っ払っててもうまくいけば好印象になるようなコマンドの出し方にしたいんですよね。

そこで，プレイヤーの思うようにはいかないようにアクション要素，ルーレットっぽいものを作ります。ここでは簡単に作れそうなものってことで，写真のように，バーの上をカーソルが移動して，うまく選べたら，うまくお話ができるというものにしてみました。やっぱり舞台がお酒を飲むバーだけに，バーを……なんてごめん，おやじギャグで。人間歳とるとこうなっちゃうんだよお。しくしく。

バニーさんのプログラミング

方針が決まったところで，実際のプログラミングにいきます。

まずは，バールーレットを作りましょう。バールーレットの外枠をBOXコマンドで描きます。BOXの中に範囲バーを作って，カーソルがこの中にあるときにうまく当たったら，うまく話せるということにします。

緑，水色，黄色，赤のバーがそれぞれ，まあよしの話題，まあまあの話題，結構よい話題，パーフェクトにいい話題であります。はずれちゃったら，おしり触るわ，からむわで最悪の行動になっていただきましょう。まさしくはずれですね。

大きさは緑のバーが一番大きくて水色はその半分の大きさ，黄色が水色の半分，赤はさらにその半分にしてみました。

気づいた人もいるかもしれませんが，このあたりのグラフィック描画やバールーレットの制御部分はいったんBASICで書い

てから，BASICコンパイラを通して，Cのソースにし，それから本体部分にくっつける，という作り方をしています。Cで作るより*BASIC*で作ったほうが楽だから，という理由もあるんですけどね。こうすると機種に依存する部分をなるべく分離して作る，ということを心がけて作るようにもなるので，結果的に他機種への移植がしやすくなるんですよ。

で，バーなんですが，酒が入ると，ろれつがまわらなくなったり，気が大きくなったり，記憶がなくなったりで，まあ，話がわけわかめ状態に突入するわけです。

さらに，酒のパラメータ,sakeが大きくなるとこのバーが狭くなり，いい話題がセレクトしにくく，はずれの行動を取る危険が大きくなるようにします。関数でいうとBar()，DispBar()のあたりがその操作を行っています。

そうそう，あとはバニーさんから見た魅力のパラメータがあったわけだけど。やさしさ，力強さ，面白さ，えっちさはそれぞれgentle，powerful，interest,sexyという名前の変数を作りました。で，バーを操作してこれらの変数の内容を増減して，バニーさんの印象を表現してます。

それで，どのコマンドを選んだか，バーの内容はどうだったかでバニーさんの変数を変更するわけですが，どうやってこの変更する内容を決めるのか結構問題。なにか式を作ってもいいんですけど……めんどくさそうなんでテーブルを作ってしまいました。これがBUNNY.Hの26行目です。int(整数)型の変数の配列でinc [] [5]っていうのがあるのがわかりますか？　こいつがそのテーブルです。{2,－10,－15,－3,－10}なんてのがならんでおりますが，こいつがpay，gentle，powerful，interest，sexyのプラスする値です。

で，一番最初の{}が「ほめる」コマンドではずしてしまった場合。続いて，「ほめる」でちょっとよい(緑)，「ほめる」で次によい(水色)，かなりよい(黄色)，パーフェクト(赤)と続いて，それから次のコマンド「笑わせる」のはずした場合……という順番で自分の話がパーフェクトだった場合まで続いてます。こうしておくと行数は増えてしまいますけども，たとえば，「ちょっとゲームのバランスが悪いな」なんてときには，このテーブルをちょちょいといじれば簡単に調整できますしね(実際，リスト掲載版はまだ，バランスがイマイチな気がするんで，自分で作り直してもらえるとありがたいんですが)。

水割りを作ってもらうときのパラメータはBUNNY.C中のdoCommand()，コマンドを入力させてそれぞれの処理に割り振る箇所で直接pay＋＝7などとしてコントロールしております。はい。

そして，バニーさんがときどきいなくなるときの処理は，isChangeBunny()が受けもっています。

また，コマンドをいろいろ試したにもかかわらずあまり内容が変わらないとバニーさんは水割りを作ります。ついでにいなくなっちゃうバニーさんもときどきいます。実際のバニーさんも話が進まなくなってくるとお酒作ったりいなくなったりしますからね。それから，コマンドをハズしまくって狼藉のかぎりを尽くした場合もバニーさんは帰ってしまいます。

でも，ゲームだからまだましなんですよ。狼藉を働くと，本当だったらバニーさんが全然こなくてさびしい思いをしなくちゃいけなくなるんだから。いや，私が狼藉を働いたわけじゃなくてだな……。

教訓4：なんか誤解されているような気がする

バニーさんとおしゃべり

それから，このゲームでとっても大きな要素をしめているのが，セリフです。

本当はグラフィック，サンプリングバリバリがいいんですが，雑誌掲載のプログラムじゃさすがに無理。潔くグラフィック，サンプリングはあきらめましょう。よってこのゲームではセリフが一番大きなウェイトを占めてます。

で，セリフを作って，いろいろとプログラムに埋め込んでいくわけでありますが，問題がいくつかあります。

それは，セリフがバレバレなことと容量です。そこでセリフデータの圧縮をやってみました。圧縮の基本は「何度も出てくる内容をうまく少ない情報で表す」ことですから，似たようなセリフを使い回す場合，多少効果があるはずです。

たとえば，「ああああああいいいうううう」とひらがなで同じような文字が続くということがわかっている場合。「続く」ということがわかっているなら，続いた回数を数字で表して「あ5い3う4」なんてやればほら，5文字分縮まったでしょ。

で，今回のプログラムでは……文字列がこんなふうになっていたのです。

「(名前)ちゃんかわいいねー☆」

「ねーちゃんかわいいのー」

「(名前)ちゃん海に行こうよ」

「ちゃん」「かわいい」などの文字が何度も出てくるのですよね。そこで，今回はキーワードのテーブルを作り，

テーブル＝"ちゃん","かわいい"

「(名前)[1][2]ねー☆」

「ねー[1][2]のー」

「(名前)[1]海に行こうよ」

[1][2]はテーブル1,2番目のキーワードという具合にして2回以上出てきそうな言葉をキーワードとしてテーブルに登録しておくことにしました。

static char ＊serifu[]={"……という配列がいろいろなコマンドのセリフが入っている配列，talk [] がセリフのテーブルです。

バニーさんと遊ぼう

ということでできたのがBUNNY.Xです。それでは実際にバニーさんとお話ししてみましょう。付録ディスクから解凍したBUNNY.C，BUNNY.Hを，

A>CC /Y /W /C bunny.c

とコマンドラインから打ち込んでコンパイルしてください。

遊び方は説明したとおりですね。さあ，君はバニーさんをお持ち帰りできるか?!

うーん，遊んでみるとバニーさんシミュレーションのつもりで作ったんだけど，思ったよりアクションゲームの雰囲気が大きくなってしまったような気がしますね。バランスを取るのにバーに頼りすぎたかな。それともやっぱりパラメータのバランスが悪いのかな。さらに，バーを選択するための矢印の移動速度もいまいち。これは，waitの値を調節してみてください。さらにゲームを遊ぶときには，手元にお酒を忘れずに(20歳未満の人はジュースで我慢してね)。

それにしてもなんかとっても恥ずい。このゲームやってると，自分がバニーさんのいる店で話してたことをテープにとられてしらふのときに聞かされているような錯覚を覚えるぞ。え？　あ，いいえっ，私はこんなことはいっさい話してませんてばっ。

教訓5：身から出たオタッキー

違うって，あれ？　この教訓ってどこかで聞いたような……ああっ，予告編と同じオチじゃないか！　まったくもう，もう大人なんだから，少しは変えないと。

教訓6：何年たっても変わらないものってあるよね……

まったくもうっ。

リスト1 BUNNY.H

```
 1: /*バニーさんの名前*/
 2: static char *BunnyName[] = {"あみ","くみこ","さちこ","ちひろ","のぶこ",
 3:                             "ふみこ","まみ","やよい","らん","わゆ"};
 4: int CurrentBunny;/*今いるバニー*/
 5: int pay ,oldpay;/*のんだだけ*/
 6: int gentle,oldgentle;/*やさしさ*/
 7: int powerful,oldpowerful; /*力強さ*/
 8: int interest,oldinterest; /*おもしろさ*/
 9: int sexy,oldsexy; /*えっちさ*/
10: int ccmd=0;/*コマンドカウンター*/
11: static char *CommandMes[]={
12:         "水割りつくってもらう","ほめる","わらわせる",
13:                             "さそう","自分の話"} ;/*コマンド名*/
14: int isPlayGame = 0;
15: int isQuitGame = 1;/*ゲームを(自分から)終りにするか*/
16: static char *talk[]={
17:    "水割り","ねーちゃん","Player","キャー","かわいい","おしり",
18:    "してるね","タイツ","ヤダ","ちゃん","バニー","ありがとうござ",
19:    "さん","お客様","さん","ちち","あたし","ぶた","けつ",
20:    "今、お作りします","ねっ☆","よかったら","はねたら","仕事が",
21:    "うふっ☆","ふふっ☆","ぬげ","ばこばこ","ははは"," ……ポッ。",
22:    "海に行","ねー","って","……","です"
23:                    };
24: static char *moku[]={
25:         "あそびにいか","ドライブしな","のみにいか","食事しにいか"};
26: static int inc[][5]={
27:  {2,-10,-15,-3,-10},{5,1,-5,0,1},{3,3,0,5,2},
28:  {2,7,3,5,10},{0,10,7,10,8},
29:  {2,-3,-10,-10,-14},{5,1,2,-3,-4},{3,3,0,2,2},
30:  {2,5,7,7,4},{0,15,10,10,6},
31:  {2,-10,-10,-15,-3},{5,-1,3,-3,1},{3,3,2,2,3},
32:  {2,7,7,5,7},{0,10,15,10,10},
33:  {2,-10,-10,-10,-10},{5,-1,-2,-0,-5},{3,3,3,2,0},
34:  {2,7,5,7,5},{0,10,10,15,10}};
35: static char *serifu[]={
36:    "[*002]*001えー*015しとんなー。¥n[@0]*003、なにするんですかぁっ!¥n",
37:    "[*002]*004*005*006。¥n[@0]え? あ、*008、*007やぶけてるぅ。¥n",
38:    "[*002]@0*009*032*010スーツにあう*031。¥n[@0]*011います¥n",
39:    "[*002]@0*009*032もてるでしょう。*004もん*031。¥n[@0]そうですかぁ? *024¥n
",
40:    "[*002]@0*012*033君の瞳はまるで星のようだ*033。¥n[@0]まぁ、*013ったら*029¥
n",
41:    "[*002]*001*015みせろやぁ!¥n[@0]*003っ!¥n",
42:    "[*002]がびちょんぶー。*017の*018ー。¥n[@0]*016*017じゃないもん*033。¥n",
43:    "[*002]となりのへいに~なんて古いやつだとお思いでしょう。¥n[@0]*025¥n",
44:    "[*002]びろ~ん☆¥n[@0]やっだーっ。きゃは*028っ☆¥n",
45:    "[*002]@0*012*033○×*033。¥n[@0]いやだわ、もうっ*029¥n",
46:    "[*002]*001*026------¥n[@0]*003ッ¥n",
47:    "[*002]*001*027せーへんかー。¥n[@0]*033ばか。¥n",
48:    "[*002]@0*009*030かない?¥n[@0]そう*031、そのうち*031。¥n",
49:    "[*002]@0*014、*030こうよ。¥n[@0]*025うん、そのときはさそ*032*031。¥n",
50:    "[*002]@0さん*033今度*033*031。¥n[@0]ええ*029¥n",
51:    "[*002]オラ、ぞう*012だぞー。¥n[@0]*003ッ!¥n",
52:    "[*002]*033*034。¥n[@0]あらそうなの。¥n",
53:    "[*002]*033なん*034。¥n[@0]*025元気出して*031。¥n",
54:    "[*002]*033*032わけなんだー。¥n[@0]おもしろーい、あ*028。¥n",
55:    "[*002]*033*032ことなん*034よ*033。¥n[@0]まぁっ*029¥n"};
56:
57: static  int     res;
58: static  int     cx;
59: static  int     cy;
60: static  int     dx;
61: static  int     bl;
62: /*以下，PCGデータは省略
```

リスト2 BUNNY.C

```
  1: /******************************************************************/
  2: /*  バニーさんシミュレーター・バニーズへようこそ☆ Ver 0.10 */
  3: /*                                              (c)de 1994     */
  4: /******************************************************************/
  5: #include <stdio.h>
  6: #include "bunny.h"
  7: #include <stdlib.h>
  8: #include <time.h>
  9: #include <conio.h>
 10: #include <ctype.h>
 11: #include <jfctype.h>
 12: #include <basic0.h>
 13: #include <graph.h>
 14: #include <sprite.h>
 15:
 16: void InitScreen( void )
 17: {
 18:   b_init();screen(0,0,1,1);
 19:   cx=30;cy=21;dx=0;
 20:   sp_init();sprite_pattern();sprite_pallet();
 21:   sp_disp(1);vpage(1);console(5,11,0);
 22:   fill(2,2,252,22,5);
 23:   symbol(9,0,"バニーズへようこそ☆",1,1,2,1,0);
 24:   symbol(7,0,"バニーズへようこそ☆",1,1,2,13,0);
 25:   box(6,29,6+23,29+23,15,'NASI');
 26:   box(38,32+4,38+200,32+4+16,15,'NASI');
 27:   sp_move(1,cx+dx,cy,0);
 28:   symbol(7,53,"■ OK",1,1,0,9,0);
 29:   symbol(128,53,"■ GOOD",1,1,0,11,0);
 30:   symbol(7,53+12,"■ BETTER",1,1,0,13,0);
 31:   symbol(128,53+12,"■ BEST",1,1,0,5,0);
 32: }
 33:
 34: void DispBar( int i )
 35: {
 36:   int  start,d1,d2,d3,d4,j;
 37:   unsigned char  dummy[32+1];
 38:   unsigned char  strtmp0[258];
 39:
 40:   j=i;res=0;d1=70-10*j;d2=d1/2;d3=d2/2;d4=d3/2;
 41:   fill(39,37,237,51,0);
 42:   fill(100,37,100+d1-1,51,9);
 43:   fill(100+d1,37,100+d1+d2-1,51,11);
 44:   fill(100+d1+d2,37,100+d1+d2+d3-1,51,13);
 45:   fill(100+d1+d2+d3,37,100+d1+d2+d3+d4-1,51,5);
 46: }
 47:
 48: int  bar(int i)
 49: {
 50:   int  start,d1,d2,d3,d4,j,d2x,k,wait=400;
 51:   unsigned char  dummy[32+1];
 52:   unsigned char  strtmp0[258];
 53:
 54:   j=i;res=0;d1=70-10*j;d2=d1/2;d3=d2/2;d4=d3/2;
 55:   DispBar(i);
 56:   d2x=0;
 57:   bl=1;
 58:   printf("なにかキーを押すとバースタート!¥nもう一度押してストップ!¥n");
 59:   b_strncpy(sizeof(dummy),dummy,b_inkeyS(strtmp0));
 60:   while ((d2x<200 & bl==1)){
 61:     sp_move(1,31+d2x,cy,0);
 62:     d2x=d2x+1;
 63:     if(d2x>199) d2x=199;
 64:     /*ウエイトループ*/
 65:     for(k=0;k<wait;k++) ;
 66:
 67:     b_strncpy(sizeof(dummy),dummy,b_inkey0(strtmp0));
 68:     if ( b_strcmp(dummy,'==',"  ") ) {
 69:       bl=0;
 70:     }
 71:   }
 72:   if ( (d2x>(62+d1+d2+d3+d4)) ) return(0);
 73:   if ( (d2x>(62+d1+d2+d3)) ) return(4);
 74:   if ( (d2x>(62+d1+d2)) ) return(3);
 75:   if ( (d2x>(62+d1)) ) return(2);
 76:   if ( (d2x>62) ) return(1);
 77:   return(0);
 78: }
 79:
 80: void  put_bunny(int i)
 81: {
 82:   if(i==0){  sp_move(4,10,32,1);beep();}
 83:   else{      sp_move(4,-15,-15,1);beep();}
 84:
 85: }
 86:
 87: int  sprite_pallet( void )
 88: {
 89:   sp_color(0,0,1);sp_color(1,18431,1);sp_color(2,42985,1);
 90:   sp_color(3,50737,1);sp_color(4,65535,1);sp_color(5,1345,1);
 91:   sp_color(6,1985,1);sp_color(7,28109,1);sp_color(8,65473,1);
 92:   sp_color(9,59393,1);sp_color(10,47105,1);sp_color(11,25407,1);
 93:   sp_color(12,47167,1);sp_color(13,63551,1);
 94:   sp_color(14,52801,1);sp_color(15,33825,1);
 95: }
 96:
 97: int  sprite_pattern( void )
 98: {
 99:   unsigned char  c[255+1];
100:
101:   movmem(_var0000,c,sizeof(c));sp_def(0,c,'NASI');
102:   movmem(_var0001,c,sizeof(c));sp_def(1,c,'NASI');
103: }
104:
105: int HitAnyKey( void )
106: /*なにかキーがおされるまでまちます*/
107: {
108:   unsigned char  dummy[32+1],strtmp0[256];
109:
110:   symbol(64,242,"HitAnyKey",1,1,0,13,0);
111:   b_strncpy(sizeof(dummy),dummy,b_inkeyS(strtmp0));
112:         fill(0,242,255,255,0);
113: }
114:
115: void KPrint(char *str)
116: {
117:   int cl;char *pStr = str;
118:
119:   cls();
120:   while(*pStr != '¥0'){
121:     cl=(int)*pStr;if(cl ==(int)'*'){
122:       int v;
123:       /*キーワードが来た*/
124:       v = ((*(pStr+1) - '0') * 100) + ((*(pStr+2) - '0') * 10) +
(*(pStr+3) - '0');
125:       printf("%s",talk[v]);
126:       pStr += 4;
127:     }else if(*pStr == '@'){
128:       int v;
129:       /*変数が来た*/
130:       v = *(pStr+1) - '0';
131:       if(v==0){
132:         printf(BunnyName[CurrentBunny]);pStr += 2;
133:       }else{
134:         printf("変数エラー¥n");pStr+=2;
135:       }
136:     }else{
137:       int c;
138:       /*単なる文字が来た*/
139:       if(iskanji(*pStr) != 0){/*2バイト文字*/
140:         c = *pStr;putchar(c);
141:         c = *(pStr + 1);putchar(c);pStr += 2;
142:       }else{/*1バイト文字*/
143:         c = *pStr; putchar(c);pStr += 1 ;
144:       }
145:     }
146:   }
147:   HitAnyKey();sp_move(1,cx,cy,0);
148: }
149:
150: void MakeDrinkDisplay(void)
151: /*水割りを作る表示*/
152: {
153:    int i,j;/*ループカウンタ*/
154:
155:   /*cls()*/;printf("[%s] %s%s%s",BunnyName[CurrentBunny],talk[0]
,talk[19],talk[20]);
156:   oldgentle = gentle;oldpowerful=powerful;oldinterest=interest;o
```

```
ldsexy=sexy;
   157:    /*水割りと作る「間」*/
   158:    for(i=0;i<4;i++){
   159:      printf(".");
   160:      for(j=0;j<5000;j++) ;/*"..."のウェイト*/
   161:    }
   162:    printf("はい、おまたせさま☆¥n");
   163:    pay += 5;
   164:    DispBar(pay/20);
   165:    HitAnyKey();
   166: }
   167:
   168: void ChangeBunny(void)
   169: /*バニーさんを交代します*/
   170: {
   171:    /*今いるバニーさんを設定します*/
   172:    CurrentBunny = rand() % 10;
   173:    /*自分に関するパラメータを初期化します*/
   174:    gentle = oldgentle = 50;/*やさしさ*/
   175:    powerful = oldpowerful = 50; /*力強さ*/
   176:    interest = oldinterest = 50; /*おもしろさ*/
   177:    sexy = oldsexy =50; /*えっちさ*/
   178:    ccmd=0;
   179:    /*ばにーのひょうじ*/
   180:    put_bunny(0);
   181:    /*自己紹介*/
   182:    printf("[%s] わたし、[%s]です。よろしく%s¥n",BunnyName[CurrentBunny],
BunnyName[CurrentBunny],talk[20]);
   183:    HitAnyKey();
   184:    /*最初に一杯水割りを作ります*/
   185:    cls();
   186:    MakeDrinkDisplay();
   187: }
   188:
   189: void Initialize(void)
   190: {
   191:    /* 実行するたびに結果が変わるように、*/
   192:    /*          random_seedの初期値をtime関数で与えます。*/
   193:    srand( (unsigned)time( NULL ) );
   194:    /*自分に関するパラメータを初期化します*/
   195:    pay = oldpay = 0; /*払ったお金*/
   196:    isQuitGame=1;/*ゲーム終了フラッグ=偽り*/
   197:     return;
   198: }
   199:
   200: void WelcomeToRoyal(void)
   201: /* 「いらっしゃいませ、バニーズへようこそ」処理 */
   202: {
   203:    cls();
   204:    printf("いらっしゃいませー☆¥n¥n ☆☆ %sズへようこそ! ☆☆¥n¥n",talk[10]);
   205:    /*バニーさんを割り振る*/
   206:    ChangeBunny();
   207: }
   208:
   209: int ReturntoHome()
   210: {
   211:    int ch;
   212:
   213:    printf("もういちどプレイしますか?(Y/N)¥n");
   214:    do{
   215:        ch = getch();
   216:        ch = toupper( ch );
   217:     } while((ch != 'Y')&&(ch != 'N'));
   218:     putch(ch);
   219:     return (ch == 'Y') ? 0 : 1;
   220: }
   221:
   222: int OneKey( void )
   223: /*ワンキーする*/
   224: {
   225:    cls();
   226:    return bar(pay/20);
   227: }
   228:
   229: void IncParm(int val)
   230: {
   231:    pay += inc[val][0];gentle += inc[val][1];powerful+=inc[val][2]
;
   232:    interest+=inc[val][3];sexy+=inc[val][4];
   233: }
   234:
   235: void homeru(int key)
   236: {
   237:    KPrint(serifu[key]);
   238:    IncParm(key);
   239:    return;
   240: }
   241:
   242: int doCommand( void )
   243: /*コマンドを実行する*/
   244: {
   245:    int ch,key=0;
   246:
   247:    cls();
   248:    printf(" 1.%s 2.ほめる 3.笑わせる ¥n 4.さそう 5.自分の話 6.帰る¥n",talk[
0]);
   249:    printf("さて,どうしようかな?(ｺﾏﾝﾄﾞ)>¥n");
   250:    do{
   251:      ch = getch();
   252:      ch = toupper( ch );
   253:    } while(( ch < '1' )||( '6'< ch ));
   254:    putch(ch);
   255:    switch(ch){
   256:      int  key;
   257:
   258:      case '1':
   259:        cls();MakeDrinkDisplay();pay+=10;ccmd=0;
   260:        break;
   261:      case '2':case '3':case '4':case '5':
   262:        key=OneKey();homeru(key+((ch-'2')*5));ccmd++;
   263:        break;
   264:      case '6':
   265:        isQuitGame = 0;
   266:        break;
   267:      default:
   268:        printf("コマンドがインプリメントされていません¥n");
   269:    }
   270: }
   271:
   272: void GoHome(void)
   273: /*うちに帰る*/
   274: {
   275:    /*一定のレベルに達してたらくどいちゃう*/
   276:    if((pay>50)&&(gentle>65)&&(powerful>65)&&(interest>65)&&(sexy>
65)){
   277:      int i;
   278:      i=rand() % 4;
   279:      cls();
   280:      printf("[%s]%s%s。あの、%s……%s%s%sない?¥n",talk[2],BunnyName[Cu
rrentBunny],talk[9],talk[21],talk[23],talk[22],moku[i]);
   281:      printf("[%s]%s¥n",BunnyName[CurrentBunny],talk[24]);
   282:      HitAnyKey();
   283:    }
   284:
   285:    cls();
   286:    if((pay>70)&&(gentle>80)&&(powerful>80)&&(interest>80)&&(sexy>
80)){/*くどきは成功したかな?*/
   287:    /*ふたりでででいく*/
   288:     printf("[%s]さーて、どこにいこうか?¥n",talk[2]);
   289:     printf("[%s]%s¥n",BunnyName[CurrentBunny],talk[24]);
   290:     printf("ふたりは仲よく帰っていきましたとさ……おめでとう!¥n");
   291:     HitAnyKey();
   292:     cls();
   293:     printf("(えっ, そのあとどうなったのかって? 作者もなったことないからわからんのだ。");
   294:     printf("編集さん, 取材ってことで連れてって～!!)¥n");
   295:     /*HitAnyKey();*/
   296:    }else{
   297:      /*一人で家に帰る*/
   298:      printf("[%s]あぁ、いい気分だなっと。",talk[2]);
   299:      printf("いい気分で、あなたはひとり、家路につきましたとさ……¥n");
   300:      HitAnyKey();
   301:      /*解説*/
   302:      cls();
   303:      if(pay<20){
   304:        printf("もっとお酒を飲んで楽しまなくちゃダメだよ。さぁ、つぎはもっと、ばーっと飲も
うね☆¥n");
   305:      }else if(gentle<51){
   306:        printf("君はやさしさに欠けていないか!?次はもっとやさしくバニーさんに接してあげよう
ね。¥n");
   307:      }else if(powerful<51){
   308:        printf("君は力強さやさしさに欠けていないか!?次はもっと自分の強さをみせなきゃだよ。
¥n");
   309:      }else if(interest<51){
   310:        printf("君っておもしろくなーい。次はもっとバニーさんをわらかしましょ☆¥n");
   311:      }else if(sexy<51){
   312:        printf("君ってかたぶつぅ。もちょっとエッチな話でもりあがろうよ!¥n");
   313:      }else{
   314:         printf("おしい、おしいっ。もうちょっとだから次もばーっといきましょ☆¥n");
   315:      }
   316:    }
   317:    printf("¥n          ***  END  ***  ¥n");
   318: }
   319:
   320: int isEnd(void)
   321: /*ゲーム終了か?*/
   322: {
   323:    if(pay>=100){
   324:      cls();
   325:      printf("[ボーイ]%s、そろそろ閉店のお時間なのですが……。¥n",talk[13]);
   326:      printf("[%s]ん……ああ……。¥n",talk[2]) ;
   327:      HitAnyKey();
   328:      GoHome();
   329:      return 0;
   330:    }
   331:    if(isQuitGame == 0){
   332:      cls();
   333:      printf("[%s]そろそろ帰ろうっと……お勘定おねがいしまーす!¥n",talk[2]);
   334:      printf("[%s]%sましたー☆¥n",BunnyName[CurrentBunny],talk[11]);
   335:      HitAnyKey();
   336:      GoHome();
   337:      return 0;
   338:    }
   339:     return 1;
   340: }
   341:
   342: int isChangeBunny()
   343: /*一定の条件になったらバニーさんをチェンジする*/
   344: {
   345:    if((ccmd>7) &&((oldgentle+10)>gentle) &&((oldgentle-10)<gentle
)){
   346:      printf("[%s]あら、グラス空だわ。¥n",BunnyName[CurrentBunny]);
   347:      MakeDrinkDisplay();
   348:      ccmd=0;
   349:      if( 1 == ( rand() % 3)) gentle = 0; /*バニーこうたい*/
   350:    }
   351:
   352:    if((gentle < 10) || (powerful < 0) || (interest < 10) || (sexy
<10)){
   353:      /*何か魅力が減ったらバニーさんがかわる……*/
   354:      cls();
   355:      printf("[%s]それでは、ごゆっくりお楽しみください☆¥n",BunnyName[CurrentB
unny]);
   356:      printf("[%s]え…あーあ、%s%s行っちゃった…¥n",talk[2],talk[10],talk[
14]);
   357:      put_bunny(1);HitAnyKey();cls();
   358:      printf("[%s]…あ、次の%s%sきた¥n",talk[2],talk[10],talk[14]);
   359:      ChangeBunny();
   360:      return 0;
   361:    }
   362:    return 1;/*バニーさんは変わらない*/
   363: }
   364:
   365: int main(int argc,char *argv[])
   366: {
   367:    InitScreen();cls();
   368:    do{
   369:      Initialize();
   370:      WelcomeToRoyal();
   371:      do{
   372:        doCommand();
   373:        if(isQuitGame != 0) isChangeBunny();
   374:      }while(isEnd() !=0 );
   375:      isPlayGame = ReturntoHome();
   376:    }while(isPlayGame == 0);
   377:    b_exit(0);
   378:    printf("[%sさんたち]おつかれさまでしたっ☆¥n",talk[10]);
   379:    return 0;
   380: }
```

降り注ぐ恐怖，再び
世紀末大戦術

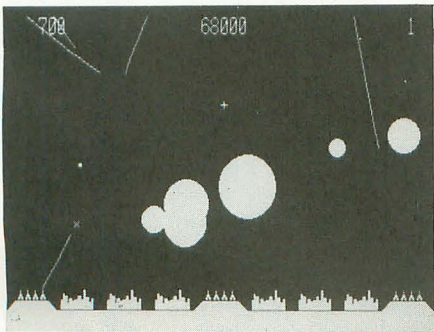
Egawano Takeshi 江川乃 誉司

懐かし度指数120%！ パワー溢れる懐ゲーです
プログラムの出来はヤバイくらいに完璧
きちんとデバッグしてから思う存分遊んでみてください

皆さん，ミサイルコマンドというゲームはご存じですよね。核戦争をテーマにするというなんともショッキングな内容，トラックボールでの操作などに誰もが心を魅了されたことと思います。

このゲームが発表されたのは，かれこれ14年も昔のことになるそうです。14年前といえば私は小学校に入って間もない頃の話で，巷のゲーセンは雰囲気が暗く，とても私の近寄るところではありませんでした(その歳で近寄る奴は誰もいねー)。ですから，私はこのゲームのアーケード版（当時はこんな呼び名もなかったんだろう）は，やったことがありません。それでも，なんだかこのゲームには変に思い入れがあり，忘れられないものがあります。

それは，それから数年後の話です。やっと世間にパソコンが普及し始め，パソコンを持っているというだけで優越感に浸れるというか，他人に自慢できるという時代がありました。当時はまだ他人だった現在の親友もそのひとりで，彼が「パソコン持ってるぜ」という話は私の耳にも早々に入ってきました。彼が購入したパソコンは，シャープのMZ-2000。グリーンディスプレイにキーボード，そしてデータレコーダまでが一体化されているという，なんとも渋いパソコンでした。

そして，彼がそのパソコンで遊ばせてくれたゲームのなかに，あのミサイルコマンドがあったわけです。当時流行っていたゲームウォッチ（懐かしすぎる）とはまったく比較にならないほどの精細な画像と，それを活かしたゲーム内容には，非常に驚かされました。

降り注ぐミサイルから都市を守れ！

おかげで，私もパソコンがむしょうに欲しくなり，しきりに親にせがんでみたものです。いま考えてみると，彼の家であのゲームをやっていなければ，私はパソコンに興味を持たずに現在に至っていたのかもしれませんね。

で今回，懐ゲー企画ということでして，なにを作ろうかいろいろ悩みましたが，やはりその例のゲームという少々（？）アナクロなゲームに挑戦することにしました。

実行方法

実行はX-BASICで行います。ただし，10月号で発表したEXEC.FNCを必要としますので，事前にBAISICに組み込んでおいてください。以下に実行に必要なファイル名を記載します。

mcm.bas	BASIC実行ファイル
mc_init.r	EXEC.FNC用実行ファイル
mc_main.r	EXEC.FNC用実行ファイル
mc_put.r	EXEC.FNC用実行ファイル
wait.r	EXEC.FNC用実行ファイル

なお，ディスクに収録してあるのは，アセンブラソースリストのみです。MAKEFILEを収録してあるので，アセンブルし直して，.rファイルを作成してください。

ちなみに付録ディスクに収録してあるリストにはバグがあります。カコミを参照して訂正しておいてください。

また，DLC.Rを使用すれば，BASICファイル1つにまとめることも可能なのですが，今回はディスク容量の関係から（テキスト化するとメモリを喰う），RファイルをBASICから直接読み出す方法を取っています。そのため，Rファイルはmcm.basと同じディレクトリに置いてください。

mcm.basは行番号なしのファイルなのでロードにはload@を使用してください。EXEC.FNCがちゃんと組み込んであれば，あとはRUNするだけで実行できます。

遊び方&基本ルール

操作にはマウスを使用します。カーソルを動かし，降り注ぐ弾道弾を先読みし，ボタンを押して迎撃ミサイルを発射してください。うまく迎撃ミサイルの爆風に巻き込めば，弾道弾を破壊することができます。

そして破壊された弾道弾はさらに誘爆を引き起こし，うまくいけば連鎖的に大量のミサイルを破壊することもできます。得点は弾道弾を破壊した高度によって変化し，より上空で破壊するほどポイントが高くなっています。

弾道弾を撃ち逃し，地表にある6つの都市が破壊されると，ゲームオーバーとなります。この場合ミサイル基地だけ残っていようとも同じことです。また，都市の数は迎撃ミサイルの性能に影響し，残り都市が少なくなると，それに応じて迎撃ミサイルの爆風半径が縮んでしまいます。

そして，一定時間の攻撃に絶え凌ぐことができればレベルクリアとなり，ボーナス得点が入ります。ボーナスは基本的に「残り都市の数×1000pts.」で求められますが，6つすべて残っているときだけは例外で，10000pts.となっています。また，レベルクリア時には，ある一定の法則に従い，都市が復旧されます。

壁作りについて

降り注ぐ大量の弾道弾を食い止めるには，単に1つひとつ狙って破壊していったのでは話になりません。このゲームでは「いかにうまく爆風の壁を作るか」が攻略のカギ

となります。

まず，弾道弾群の輪郭をつかんでください。そして，少し手前の位置で，その輪郭をなぞるようにして素早くカーソルを移動させるのです。このとき，うまいこと等間隔に迎撃ミサイルを発射していけば，強力な爆風壁の出来上がりです。そのあとの豪快な誘爆を楽しみましょう。

しかし，この壁作りに関して注意してほしい点に，迎撃ミサイルの同時発射可能数があります。迎撃ミサイルを発射できる基地は，左右真ん中と3つありますが，それぞれ4台の砲台で1組となっています。残弾数には制限ありませんが，同時に発射できるミサイルは1砲台につき1つまでとなっており，発射されたミサイルの爆風が完全に消えるまでは，次のミサイルを発射することができません。

このゲームはカーソルに一番近い基地と砲台を自動的に選んで発射してくれるという親切設計になっているのですが，あまり同時に撃ちすぎて発射可能砲台が少なくなってくると，カーソルからどんなに離れた砲台であろうと発射可能でさえあればかまわず発射してしいます。つまりその砲台は次のミサイルが発射可能になるまで，非常に時間がかかってしまうことになります。それを防ぐためには，各基地ごとに担当エリアを分けて，同時に発射するミサイルはそれぞれエリアで4発までと，心がけておくとよいでしょう。

これを踏まえたうえで壁作りを行えば，もうばっちりです。より高レベルを目指してがんばってください。

弾道弾種について

敵弾道弾にはいくつかの種類があるので紹介しておきます。

A．通常型

単に等速度で落下し，並みの爆風を起こす普通の弾道弾です。出現の割合は，このタイプが一番多くなっています。

B．分散型

途中まではAタイプよりもゆっくり落下してくるのですが，ある地点で3つ以上の弾頭に分散し，同時に速度を上げて襲ってくるというトリッキーな弾道弾です。しかし，うまいこと分散前に破壊さえすれば巨大な爆発を引き起こしますので，多くの誘爆を期待できます。

C．大爆発型

動きとしてはBタイプと同じなのですが，Bタイプでは途中で分散するところを，なにを血迷ったのか分散せずに，そのまま突っ込んでくるという弾道弾です。恐ろしいのはその破壊力で，その爆風は落下地点のまわりの建物まで巻き込むほど超巨大です。しかし，これも破壊さえすれば非常に多くの誘爆を期待できるので，逆に利用したいところです。

B,Cタイプを攻略するには常に上空の弾道弾に気を配ることが大切です。結構微妙ではありますが，なんだか落ちてくるのが遅いなぁと思える弾道弾は，速やかに葬ってやりましょう。

技術的な話

このゲームでは見てのとおりグラフィックに線や円を描いて画面を構成しています。そこで見てもらいたいのが，重ね合わせ処理です。いわなければおそらく気づいてもらえないようなさりげない処理なのですが，読者の皆さんは，画面を見て気づいたでしょうか。

たとえば弾道は描画されたらされたで，そのままにして置くわけではなくて，着弾したときに消去する必要があります。消去の方法としては，黒で上から塗り潰すということが考えられると思います。しかし，弾道が交差している部分を考えてみてください。交差している一方の弾道を黒で上描きした場合，もう一方の弾道はその交差していた部分で切れて表示されてしまいます。また，爆風描画のときは，爆風の収縮時に円を徐々に消去していくのですが，こちらもし黒で上描きする方法を取った場合，たとえば2つの爆風が重なっている部分があったとしたら，それらは互いに相手の円を消し合い，これも表示がおかしくなってしまいます。

この描画の重なった部分を，お互いに影響を与えることなく表示させる処理が，この重ね合わせ処理となります。一見難しそうな処理だと思われますが，その方法は実に単純なものです。

まずは0以外のパレットを同一色で埋めておきます。あとは，描画時にひと工夫するだけなのですが，もうおわかりでしょう。つまり，前述の方法のように「描くときは1の色の点を打ち，消すときは0の点を打つ」といった，単に用途に対応した色を書き込むのではなく，「描くときはその座標の色に1を足し，消すときは1を引く」というように重なれば重なっただけ，色を加算していくようにすればよいわけです。ということは，グラフィック上では重なり具合によって，別の色が指定されていることになるのですが，パレットには同一色が設定されているため，見た目にまったく同じ色となります。

これならたとえ交差した弾道を消したとしても，交差部分はより多く色が加算されているので，減算後も消去されずに残ります。よって，爆風の重複部もまったく破綻せず描画することができるわけです。

さらに，パレットの設定次第ではもっと面白いことができます。すべてのパレットに同一色を設定するのではなく，たとえば

世紀末大戦術のデバッグ

非常に申しわけありませんが，今回の付録ディスクに収録した「世紀末大戦術」にいくつかのバグが発覚しました。ソースリストを付録ディスクから解凍したら，以下の箇所を変更してください。

●mcm.basの変更点

パーフェクトを取ると終わらなくなるバグを修繕。

69，72，81行（LOAD@で読み込み時の690，720，810行），

```
690 if r=6 then r=10
720 s= s + r *1000
810 center (7," BONUS "+n$(r)+"000pts.")
```

を次のように変更

```
690 if r<6 then z=r else z=10
720 s= s + z *1000
810 center (7," BONUS "+n$(z)+"000pts.")
```

●mc_main.rの変更点

レベル調整を補正するようにします。

558行から560行までを消去し，次のリストを挿入。

```
558:         move.w   lv(a3),d0
559:         lsr.w    #1,d0
560:         cmpi.w   #9,d0
561:         ble      lvok
562:         moveq.l  #9,d0
563: lvok:
564:         add.w    d0,d6
565:         subq.w   #7,d6
566:         bge      disokp
567:         clr.w    d6
568:         bra      disok
569: disokp:
570:         cmpi.w   #6,d6
571:         ble      disok
572:         moveq.l  #6,d6
```

201，311行，

```
201:         addq.w   #7,d0
311:         bge      vdisp_wait
```

を次のように変更。

```
201:         addq.w   #8,d0
311:         beq      vdisp_wait
```

上半分のパレットコードに黒を設定するようにしてみます。そして都市などの描画には，もう1つでも加算したら黒に変わる色を使用するのです。これに爆風などがかぶさると，その重複部の表示は反転されるので，爆風のなかに都市のシルエットが浮かび上がるといった，いかにも爆風で焼かれているような表現ができます。

また，工夫さえすれば多発色描画も可能でしたが，私にはMZ-2000の単色グリーン画像の印象が強いので，変に色をつけるよりはこのほうが渋くてよいのではないかと思い，単色表示にしてみました。なお19行目のRGB関数の値を変えることに，その色を変更できるようになっています。

重ね合わせについてはこれくらいですが，今回グラフィックを使用するに当たって，IOCSの遅さを非常に痛感しました。制作に与えられた期間は1週間ほどしかなかったため，最初は円描画はIOCSを使って楽をしようと考えていたのですが，使ってみたらとんでもなく遅い。半径30弱の円を1/60秒間に，たったの2個程度（10MHz時）しか描けないのです。このゲームにおいては大量の円描画を高速に処理できなければ，爆風の壁も作れないわけでして，なんの面白味もなくなってしまいます。ましてや2個といったら話にもなりません。

また，この円描画では半径1ずつずらした円を並べていっても，円と円の隙間が埋まらず，爆風がスカスカになってしまうので，結局自分で作るはめになりました。その際，村田氏の「X68000マシン語プログラミング　グラフィックス編」を参考にさせて頂きました。そのおかげで，数十個を余裕で描画できる円描画ルーチンをあっという間に作ることができました。このようなすばらしい本を世に送りだしてくれた村田氏に，この場を借りて深く感謝したいと思います。

おまけ

今回の付録ディスクには，10月号で紹介したEXEC.FNCのサンプルゲーム「SHOCK TROOPS！」のトレースモードつきバージョンSHOCK2.BASが収録されています。私のトレースデータもありますので，暇があったら見てやってください。

なお，実行には今回のSHOCK2.BAS以外に10月号の付録ディスク収録の以下のファイルが必要となります。

ENEMY.DAT	敵アルゴリズムデータ
EXP.PCM	効果音PCMデータ
HIT.PCM	効果音PCMデータ
IGN.PCM	効果音PCMデータ
SHOT.PCM	効果音PCMデータ

これらをSHOCK2.BASと同ディレクトリに置いて，BASIC上からRUNしてください。

トレースデータはベストタイムを出すとTRACE.DATというファイルに記録されます。このファイルはプログラム実行時にバッファに読み込まれ，タイトル画面でジョイスティックの十字キーを押すとトレースを実行することができます。

ちなみに私のトレースデータはOMAKE.DATというファイルに入っています。トレースを見るためにはファイル名をTRACE.DATにリネームしてください。

しかし，今回は死ぬほどきつかった。私は，時間がないときでも，最低限はまともに作らないと気が済まないたちなのですが，そのへんのノルマはなんとかこなせたのではないかと思います。それでも，まだまだやり残したと思えることは無数にあり，結局あまり満足できるものに仕上がらなかったのは，少々残念に思います。

今度このような機会があれば，ぜひじっくり時間をかけたものをお届けしたいと考えています。

リスト1　mc circle.s

```
 1:          .xdef   mc_circle
 2:
 3:
 4:          .offset 0
 5:
 6: xyadr:   .ds.l   1
 7: r:       .ds.w   1
 8: col:     .ds.w   1
 9:
10:
11:          .text
12:          .even
13:
14: *----------------------------------------
15:
16: mc_circle:
17:          link    a6,#0
18:          movem.l d0-d7/a0-a3,-(sp)
19:          movea.l 8(a6),a1
20:
21:          move.l  (a1)+,a0
22:          subq.w  #2,a0
23:          movem.w (a1),d2/d7
24:          andi.w  #255,d7
25:          move.w  d7,d6
26:          swap.w  d7
27:          move.w  d6,d7
28:
29:
30:          move.l  d2,d0
31:          moveq.l #10,d6
32:          lsl.l   d6,d0
33:          lea.l   0(a0,d0.l),a2
34:          neg.l   d0
35:          lea.l   0(a0,d0.l),a3
36:
37:          move.l  d2,d0
38:          moveq.l #0,d1
39:
40:          add.w   d2,d2
41:          adda.l  d2,a0
42:          movea.l a0,a1
43:
44:          move.l  d2,d4
45:          neg.l   d4
46:          addq.l  #1,d4
47:
48:          add.l   d2,d2
49:          moveq.l #2,d3
50:
51:          move.l  d2,d5
52:          neg.l   d5
53:          moveq.l #0,d6
54:
55:
56: loop:
57:          add.l   d7,(a0)
58:          add.l   d7,(a1)
59:          add.l   d7,(a2)
60:          add.l   d7,(a3)
61:          add.l   d7,2(a0,d5.l)
62:          add.l   d7,2(a1,d5.l)
63:          add.l   d7,2(a2,d6.l)
64:          add.l   d7,2(a3,d6.l)
65:
66:          add.l   d3,d4
67:          bmi     vmove
68:
69: dmove:
70:          subq.w  #1,d0
71:          subq.l  #4,d2
72:          sub.l   d2,d4
73:
74:          subq.w  #2,a0
75:          subq.w  #2,a1
76:          lea.l   -1024(a2),a2
77:          lea.l   1024(a3),a3
78:          addq.l  #4,d5
79:
80: vmove:
81:          addq.w  #1,d1
82:          addq.l  #4,d3
83:
84:          lea.l   1024(a0),a0
85:          lea.l   -1024(a1),a1
86:          addq.w  #2,a2
87:          addq.w  #2,a3
88:          subq.l  #4,d6
89:
90:          cmp.w   d1,d0
91:          bge     loop
92:
93:
94:          movem.l (sp)+,d0-d7/a0-a3
95:          unlk    a6
96:          rts
97:
98:
99:          .end
```

リスト2 mc line.s

```
 1:         .xdef   mc_line
 2:
 3:
 4:         .offset 0
 5:
 6: xyadr:  .ds.l   1
 7: l:      .ds.w   1           *d0
 8: col:    .ds.w   1           *d1
 9: e:      .ds.w   1           *d2
10: dx:     .ds.w   1           *d3
11: dy:     .ds.w   1           *d4
12:
13:
14:         .text
15:         .even
16:
17: *---------------------------------------
18:
19: mc_line:
20:         link    a6,#0
21:         movem.l d0-d6/a0-a1,-(sp)
22:         movea.l 8(a6),a1
23:         move.l  (a1)+,a0
24:         movem.w (a1),d0-d4
25:         andi.w  #255,d1
26:
27:
28:         move.w  #$400,d5
29:         add.w   d4,d4               * 2dy
30:         bge     line1
31:         neg.w   d4
32:         neg.w   d5
33: line1:
34:         moveq.l #2,d6
35:         add.w   d3,d3               * 2dx
36:         bge     line2
37:         neg.w   d3
38:         neg.w   d6
39: line2:
40:         cmp.w   d3,d4
41:         bgt     yline
42:
43:
44: xline:
45:         add.w   d1,(a0)             *pset
46:         adda.w  d6,a0
47:         add.w   d4,d2
48:         bmi     xline1
49:         adda.w  d5,a0
50:         sub.w   d3,d2
51: xline1:
52:         dbra    d0,xline
53:         bra     done
54:
55:
56: yline:
57:         add.w   d1,(a0)             *pset
58:         adda.w  d5,a0
59:         add.w   d3,d2
60:         bmi     yline1
61:         adda.w  d6,a0
62:         sub.w   d4,d2
63: yline1:
64:         dbra    d0,yline
65:
66:
67: done:
68:         move.l  a0,-(a1)
69:         move.w  d2,8(a1)
70:         movem.l (sp)+,d0-d6/a0-a1
71:         unlk    a6
72:         rts
73:
74:
75:         .end
```

リスト3 mc put.s

```
  1:        .include        iocscall.mac
  2:
  3:
  4:        .offset 0
  5:
  6: xyadr: .ds.l   1
  7: pdadr: .ds.l   1
  8: on:    .ds.w   1
  9:
 10:
 11:        .text
 12:        .even
 13:
 14: *---------------------------------------
 15:
 16: mc_put:
 17:        lea.l   0.w,a1
 18:        IOCS    _B_SUPER
 19:        addq.l  #1,d0
 20:        andi.b  #%11111110,d0
 21:        move.l  d0,-(sp)
 22:
 23:        move.w  6(a2),-(sp)
 24:        lea.l   city(pc),a0
 25:        move.l  (a2),d0
 26:        move.l  d0,d1
 27:        andi.b  #%11,d1
 28:        bne     put_city
 29:        lea.l   base(pc),a0
 30: put_city:
 31:        pea.l   (a0)
 32:
 33:        lea.l   $c10080+$400*220,a0
 34:        mulu.w  #84,d0
 35:        pea.l   16(a0,d0.w)
 36:
 37:        pea.l   (sp)
 38:        jsr     mput(pc)
 39:        lea.l   14(sp),sp
 40:
 41:        move.l  (sp)+,a1
 42:        IOCS    _B_SUPER
 43:        moveq.l #0,d0
 44:        rts
 45:
 46: mput:
 47:        link    a6,#0
 48:        movem.l d0-d7/a0-a2,-(sp)
 49:        movea.l 8(a6),a2
 50:        move.l  (a2)+,a0
 51:        move.l  (a2)+,a1
 52:        move.w  #128,d1
 53:        move.w  (a2),d0
 54:        bne     mput1
 55:        not.w   d1
 56:
 57: mput1:
 58:        move.l  (a1)+,d2
 59:        beq     mput_done
 60:        moveq.l #31,d5
 61:        tst.w   d0
 62:        bne     mput4
 63: mput2:
 64:        add.l   d2,d2
 65:        bcc     mput3
 66:        and.w   d1,(a0)+
 67:        dbra    d5,mput2
 68:        bra     mput6
 69: mput3:
 70:        addq.w  #2,a0
 71:        dbra    d5,mput2
 72:        bra     mput6
 73:
 74: mput4:
 75:        add.l   d2,d2
 76:        bcc     mput5
 77:        or.w    d1,(a0)+
 78:        dbra    d5,mput4
 79:        bra     mput6
 80: mput5:
 81:        addq.w  #2,a0
 82:        dbra    d5,mput4
 83: mput6:
 84:        lea.l   $3c0(a0),a0
 85:        bra     mput1
 86:
 87: mput_done:
 88:        movem.l (sp)+,d0-d7/a0-a2
 89:        unlk    a6
 90:        rts
 91:
 92: city:
 93:        .dc.l   %00000000000000000000001100000000
 94:        .dc.l   %00000000000000000000001100000000
 95:        .dc.l   %00000000000000000000001100000000
 96:        .dc.l   %00011000000000000000001100000000
 97:        .dc.l   %00011000000000000000001100000000
 98:        .dc.l   %00011111000000000111001100000000
 99:        .dc.l   %00011111000000000111001111100000
100:        .dc.l   %00011110111000001110011111100110
101:        .dc.l   %01111110111001111111111111100110
102:        .dc.l   %01111110111111111111111111100110
103:        .dc.l   %01111110111111111111111111111110
104:        .dc.l   %01111111111111111111111111111110
105:        .dc.l   %01111111111111111111111111111110
106:        .dc.l   %01111111111111111111111111111110
107:        .dc.l   %01111111111111111111111111111110
108:        .dc.l   %01111111111111111111111111111110,0
109:
110: base:
111:        .dc.l   %00000100000010000001000000100000
112:        .dc.l   %00000100000010000001000000100000
113:        .dc.l   %00000100000010000001000000100000
114:        .dc.l   %00000100000010000001000000100000
115:        .dc.l   %00001110000111000011100001110000
116:        .dc.l   %00001110000111000011100001110000
117:        .dc.l   %00011011001101100110110011011000
118:        .dc.l   %00011011001101100110110011011000,0
119:
120:        .end
```

▶会社をやめてから，かなり(2年)になります。免許も取ったし，そろそろ就職しないと。
林　秀樹(27)兵庫県

リスト4　mcm.bas

```
  10 /*---------------------------- 初期設定
  20 char mc_init(4095),mc_put(4095),mc_main(4095),wait(4095)
  30 char work1(4095),work2(4095),clr(4095)
  40 int i,z,r,lv,s,hs
  50 str n$(99)
  60 for i=0 to 9
  70 n$(i)=chr$(&H82)+chr$(&H4F+i)
  80 next
  90 for i=10 to 99
 100 n$(i)=n$(i¥10)+n$(i mod 10)
 110 next
 120 fread(mc_init,4096,fopen("mc_init.r","r"))
 130 fread(mc_put,4096,fopen("mc_put.r","r"))
 140 fread(mc_main,4096,fopen("mc_main.r","r"))
 150 fread(wait,4096,fopen("wait.r","r"))
 160 fcloseall()
 170 cls
 180 pokel(_svp(0),_arp(work1),_arp(work2))
 190 exec(mc_init,rgb(0,31,0),0)
 200 vpage(0)
 210 ground()
 220 city()
 230 print spc(6);"0";spc(15);"68000";spc(18);"0"
 240 /*---------------------------- タイトル
 250 while 1
 260 console 1,15,0
 270 center(5,"世紀末　大戦術")
 280 center(10,"(C)1994 ORE ")
 290 vpage(3)
 300 repeat
 310 until msbtn(1,0,16)>=1
 320 /*---------------------------- 準備1
 330 cls
 340 console 0,15,0
 350 print"       0"
 360 lv=1
 370 /*_svp(8).w
 380 /*レベル
 390 /*面クリアまでの残り時間
 400 /*残り都市数
 410 /*残り基地数
 420 /*地破壊状況
 430 /*スコア（上位ワード）
 440 /*スコア（下位ワード）
 450 poke(_svp(8),lv,60*30,6,3,0,0,0)
 460 /*---------------------------- 準備2
 470 while 1
 480 /*move(_arp(clr),_arp(work1),4096)
 490 /*move(_arp(clr),_arp(work2),4096)
 500 city()
 510 locate 44,0
 520 exec(wait,60)
 530 print using"##";lv
 540 center(6,"LEVEL "+n$(lv))
 550 exec(wait,90)
 560 center(6,"   READY   ")
 570 exec(wait,90)
 580 center(6,"           ")
 590 setmspos(192,128)
 600 poke(_arp(work1),1)
 610 /*---------------------------- メイン
 620 repeat
 630 r=exec(mc_main)
 640 until r
 650 move(_arp(clr),_arp(work1),4096)
 660 move(_arp(clr),_arp(work2),4096)
 670 /*---------------------------- レベルクリア
 680 if r>0 then {
 690 if r<6 then z=r else z=10
 700 s=peekl(_svp(18))
 710 hs=peekl(_svp(22))
 720 s=s+z*1000
 730 if s>hs then hs=s
 740 pokel(_svp(18),s,hs)
 750 center(5,"OVERCAME!")
 760 locate 0,0
 770 exec(wait,90)
 780 print using"#######";s;
 790 locate 20,0
 800 print using"#######";hs;
 810 center(7,"BONUS "+n$(z)+"000pts.")
 820 exec(wait,90)
 830 console 1,20,0
 840 cls
 850 console 0,20,0
 860 exec(wait,30)
 870 lv=lv+1
 880 z=peek(_svp(16))
 890 if (z and &B100) then {
 900 z=z and &B11111011
 910 r=r+1
 920 } else if (z and &B1010) then {
 930 r=r+1
 940 if (z and &B1010)=&B1010 then r=r+1
 950 z=z and &B11110101
 960 }
 970 if (z and &B1000000) then {
 980 z=z and &B10111111
 990 r=r+1
1000 } else if (z and &B10100000) then {
1010 r=r+1
1020 if (z and &B10100000)=&B10100000 then r=r+1
1030 z=z and &B101011111
1040 }
1050 z=z and &B11101110
1060 poke(_svp(8),lv,60*30,r,3,z)
1070 continue
1080 }
1090 /*---------------------------- GAME OVER
1100 color 1
1110 center(7,"GAME OVER")
1120 color 3
1130 exec(wait,180)
1140 center(7,"                 ")
1150 vpage(0)
1160 sp_off()
1170 apage(1)
1180 wipe()
1190 apage(0)
1200 wipe()
1210 ground()
1220 poke(_svp(16),0)
1230 city()
1240 break
1250 endwhile
1260 endwhile
1270 end
1280 /*---------------------------- 画面中央に文字を表示
1290 func center(y;int,s;str)
1300 locate 24-len(s)/2,y
1310 print s
1320 endfunc
1330 /*---------------------------- 地面を描く
1340 func ground()
1350 fill(64,301,448,319,1)
1360 for i=0 to 2
1370 z=i*168
1380 line(64+z,301,72+z,293,1)
1390 line(72+z,293,103+z,293,1)
1400 line(103+z,293,111+z,301,1)
1410 paint(72+z,294,1)
1420 next
1430 endfunc
1440 /*---------------------------- オブジェクトを描く
1450 func city()
1460 z=peek(_svp(16))
1470 for i=0 to 8
1480 if (z and 1)=0 then {
1490 exec(mc_put,i,1)
1500 }
1510 z=z shr 1
1520 next
1530 endfunc
```

垂直帰線期間待ち外部関数

ちょっとスペースが余ったので，ここで「北斗の男」で使われているX-BASICで垂直帰線期間待ちをする外部関数を紹介します。とりあえず，リストをアセンブル，リンクしてから，拡張子を.FNCとして，X-BASICに組み込むだけでOKです。

すると「v_disp()」という関数が拡張されます。この関数を実行すると，垂直帰線期間を待ってリターンしてきます。引数はありません。

リスト

```
 1: *
 2: *  垂直帰線期間待ち
 3: *
 4:
 5:        .include        iocscall.mac
 6:        .include        doscall.mac
 7:
 8:        .even
 9:
10:        .xdef   _v_disp
11:
12:
13:        dc.l    ret
14:        dc.l    ret
15:        dc.l    ret
16:        dc.l    ret
17:        dc.l    ret
18:        dc.l    ret
19:        dc.l    ret
20:        dc.l    ret
21:        dc.l    name
22:        dc.l    para
23:        dc.l    entry
24:        ds.l    5
25:
26: ret:
27:        rts
28:
29: entry:
30:        dc.l    _v_disp
31:
32: _v_disp:
33:        move.l  #0,a1
34:        IOCS    _B_SUPER
35:        move.l  d0,sspbuf
36:
37:        lea.l   $e88001,a6
38:        moveq.l #4,d7
39: waitdisp1:
40:        btst.b  d7,(a6)
41:        beq     waitdisp1
42: waitdisp2:
43:        btst.b  d7,(a6)
44:        bne     waitdisp2
45:
46:        move.l  sspbuf,a1
47:        IOCS    _B_SUPER
48:        rts
49:
50:        .even
51:
52: sspbuf:
53:        dc.l    00000000
54: para:
55:        dc.l    para_m
56:        dc.l    para_m
57: para_m:
58:        dc.w    $ffff
59: name:
60:        dc.b    'v_disp',00
61:        dc.b    00
```

▶夏の暑い頃は，ドライブ0のフロッピーをはき出さなくなり困っていましたが，秋になって正常に作動しております。ちなみに過去に2回，修理に出しています。
竹内　清志(26)東京都

あの名作が再び！

復刻版「北斗の男」

Hamazaki Masaya 浜崎 正哉

いまここに往年の名作「北斗の男」が復活をとげました
完全移植とはいかなかったようですが，オリジナルの雰囲気はそのまま再現
キャラグラで構成される味のあるキャラクターたちを堪能してください

あの名作がここに復活！

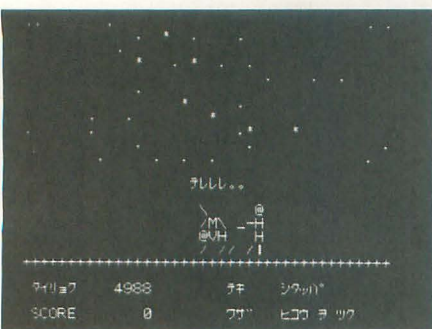
なんでも，今月号でOh!Xが7周年。で特別企画として「懐ゲー制作工房」をやるということなので，日頃ゲームばかり作っている僕としては，ぜひとも参加せねばならない企画です。さっそく頼まれてもいないうちから首を突っ込んで「懐ゲーといったらねえ」と考えているうちに浮かんだのが，あの「北斗の男」でした。

実際に遊んだことはないのですが，8年前のOh!MZ1986年10月号に掲載され，ずいぶんと話題を呼んだ作品です。当時，あまりに巨大なプログラムなため打ち込む気力もなく，ただ「面白そうだなあ」と眺めていただけでした。

そこで，あの名作を遊んでみたい，という思いにかられ移植作業に手を染めてしまったのです（しかし，1時間後には泣きが入りましたが）。

遊び方

付録ディスクから解凍したらX-BASICを立ち上げ，RUNするだけです。なるべくならコンパイルしてほしいので，後述の「コンパイルのについて」にしたがってコンパイルしてください。

ゲームが立ち上がると，どこかで見たようなストーリーが表示され，なにかキーを押すか，しばらく放っておくと操作説明の画面になります。読みにくい半角カタカナの説明文を理解したら，なんでもいいからキーを押してください。するとゲームが始まります。

7つの傷を持つ男が帰ってきた！

画面の右側にいるのがプレイヤーが操るケンタロウです。ケンタロウは，
・4，6キー……左右に移動します
・8キー……前ジャンプ
・2キー……後ろジャンプ
・Cキー……ニシシンクウハ（スペードと戦うときに必要）
・Z，Xキー……技の選択
・スペースキー……敵に技を繰り出します
以上のキーで操作できます。

そして，左側に現れた敵をバシバシと倒していくのですが，ケンタロウの体力が0になってしまうとゲームオーバーです。もう一度遊びたいときにはRキーを押してください。そのほかのキーを押すとゲームは終了します。

ケンタロウの技について

ケンタロウが使える技には，
・セイケンヅキ
・アシゲリ
・ヒコウテツク（3種類）
・ホクトザンカイケン
・ホクトヒャクレツケン
・コウシュハガンケン
・ホクトカイコツケン
・ホクトジュウハザン
以上8種類があります。

これらの技に必要な間合いは違いますので，タイミングをしっかり覚えましょう。敵によって効かない技もありますが，決まりさえすれば相手を一撃で倒すことができます（セイケンヅキとアシゲリを除く）。

そのときの敵の情けないセリフがいくつか用意されていますので，楽しんでください。

あと，X68000版では前ジャンプするときにキックを繰り出します。これは，オリジナルにはたぶんなかったものですが，思い切ってつけてしまいました。相手に向かって跳び込むときに使えるので，あってもかまわないでしょう。

敵は10人

敵はオリジナルと同じく10人登場します。さまざまな攻撃でケンタロウを苦しめますので，よく敵の攻撃パターンを見定め（といってもほとんど乱数だけど）必殺技をお見舞いしてやりましょう。

・シタッパ……セイケンヅキと体当たりで攻撃してきます。

・スペード……ボウガンを持っていて，ときどき矢を放ってきます。矢はてろてろと飛んでくるので，前ジャンプでかわすか，すぐに“C”キーを押し，ニシシンクウハで防ぐようにしてください（かなりタイミングはシビアだけど）。矢が1発でも当たるとケンタロウは死にます。

・クラブ……棒使いの達人。離れたところから長い棒で攻撃してきます。

・ハート……かの有名なハート様。わけのわからない武器を持ち，「イテェヨウ」のセリフを連発して狂ったように襲いかかってきます。特異体質でケンタロウのホクトシンケンをもってしても有効な技がほとんどありません。

・ジャギ……ホクトシンケンとナントセイケンを使い分けますが，どっちつかずの修業をしてきたためにそれほど強力ではありません。しかし，必殺のホクトラカンゲキには注意してください。

・ユダ……ナント六聖拳のひとり。ナントコウカクケンは攻撃範囲が広いので注意しましょう。

・レイ……ユダ同様，ナント六聖拳のひとり。ナントスイチョウケンで相手を切り刻むので，容易に近づけず，かなり手強い敵といえます。
・ラオウ……とにかく強い。ケンタロウの技もほとんど通用しません。特にシンケッシュウを突かれると，体力は減りませんが，残り3日（ゲーム中では3分）の命となってしまいます。

あと，オリジナルでは，これらの敵がランダムに出現してきたのですが，X68000版では順番に出現してきます。

コンパイルについて

あまりたいした処理はしていないにもかかわらず，インタプリタ上で実行するにはかなり遅いものとなってしまいました。多分，10MHzマシンでは遊ぶに堪えない速度だと思いますので，コンパイルすることをお勧めします。

で，コンパイルするにあたっては，4行目のcompの値を1に設定してからコンパイルするようにしてください（インタプリタの場合は0）。

なお，コンパイルはGCCでのみチェックを行いました。そのときつけたオプションは，

-O -fstrength-reduce -finline-functions -fomit-frame-pointer -fwritable-strings

としました。

あと，ウェイトに垂直帰線期間を検出する関数v_disp()を使用しています。これは，61ページにソースリストを掲載してあります。各自，外部関数を作成するか，似たような関数があればそちらを使用してください。どうにもならない場合は，空ループに書き換えてください。

移植度について

一応，リストにはひととおり目を通して，だいたいのところは再現できたと思います。しかし，リストを読み切れない部分も多々あり，多少アレンジを加えてしまいました。元のリストが，とんでもなくスパゲティなプログラムで，あちこちに散らばる変数が，どのような意味をもっているのか完全に把握しきれなかったのが原因です（要するに僕の力不足なんだけど）。

そして，MZ-2000用のリストをそのまま移植したため，画面は単色のみの寂しい画面になってしまいました。あと，時間の関係上，サウンドまで手が回りませんでした。それらしいMUSIC文をコメント化してありますので，暇がある人はチャレンジしてみるといいでしょう。

さらに，X68000ではグラフィックキャラクタが使えないため，オリジナルとはかなりパターンが違っています。最初はスプライトを使って完全移植を目指そうとしたのですが，とりあえず動かすだけでもひと苦労な状態になってしまい，これ以上手を入れようとするととても間に合わなくなってしまったのです。

こうしてみていくと，かなりオリジナルと違うものになってしまいましたね（反省）。しかし，画面の雰囲気，キャラクターの動きは再現していると思いますので，一挙一動に笑いをかみしめながら遊べることでしょう。

それにしても，出来上がったリストのでかいこと。はっきりいって，X-BASICでここまで大きなプログラムを組んだのは初めてです。オリジナルのリストも，BASICのフリーエリアを使い切るほどでしたから当然なんですけどね。打ち込むだけでも疲れましたから。

では，ホクトシンケンの真の伝承者となるためにがんばってください。

リスト

```
  1: /*
  2: /* 「北斗の男」for X68000
  3: /*
  4: int comp=0      /*コンパイルスイッチ0:インタプリタ，1:コンパイル
  5: dim str na(10)={ "ﾀﾞﾐｰ","ｼﾀｯﾊﾟ  ","ｽﾍﾟｰﾄﾞ","ﾀﾞｲﾔ   ","ｸﾗﾌﾞ   ",
  6:         "ﾊｰﾄ   ","ﾚｲ    ","ｼﾞｬｷ   ","ﾕﾀﾞ    ","ｼｭｳ   ","ﾗｵｳ   " }
  7: dim str wa(8)={ "ﾀﾞﾐｰ","ｾｲｹﾝﾂﾞｷ       ","ｱｼｹﾞﾘ          ","ﾋｺｳ ｦ ﾂｸ
 ",
  8:                   "ﾎｸﾄ ｻﾞﾝｶｲｹﾝ  ","ﾎｸﾄ ﾋｬｸﾚﾂｹﾝ  ","ｺｳｼｭ ﾊｶﾞﾝｹﾝ  ",
  9:                   "ﾎｸﾄ ｶｲｺｳｹﾝ   ","ﾎｸﾄ ｼﾞｭｳﾊｻﾞﾝ" }
 10: dim str hh(5)={ "ﾀﾞﾐｰ","ﾅﾆ!!!","ﾊﾞｶﾅ!","ﾅﾝﾀﾞﾄ","ｳｫｯ!!","ｷｶﾅｲ!" }
 11: dim str w(3)={ "ﾀﾞﾐｰ","ﾄｳｲ","ｾﾂﾘ","ｲｶｸ" }
 12: dim str zs(4)={ "ﾀﾞﾐｰ","ﾍｯﾍｯﾍ","7ﾂﾉ ｷｽﾞﾉ","ｵﾄｺﾉ ｸﾋﾞﾊ","ｵﾚｻﾏ ｶﾞﾓﾗ
ｯﾀ!!" }
 13: dim str sh(11)={ "ﾀﾞﾐｰ","ﾋﾃﾞﾌﾞ!","ﾜﾊﾞﾗ!","ﾀﾜﾊﾞ!","ｱﾍﾞｼ!","ﾊｶﾞｶﾞ"
,
 14:                    "ｴﾛﾊﾞ!","ﾁﾆｬ!!","ｱﾛ!!!","ﾗｲﾚｯ!","ﾋｴ!!!","ﾍﾚｯ!!"
 }
 15: dim str win(4)={ ".......","ﾌｯ.....","ｱｸﾄｳﾆ ﾎﾞﾋｮｳﾊｲﾗﾅｲ",
 16:                   "ｷｻﾏｼﾞｼﾝｦ ｼﾗﾅｲﾉﾊ ｷｻﾏﾀﾞｹﾀﾞ..","ﾊﾊﾊﾊﾊ" }
 17: dim int q(8)
 18: /*ワザのスコア
 19: dim int sw(7)={ 10,30,100,150,200,270,200,180 }
 20: dim str t1(7),t2(14)
 21: dim str k(7)[255],s(10)[255],bo(3),bk(3)[255]
 22: str ju="                    "
 23: str k1[255],k2[255],k3[255],k4[255]
 24: str k5[255],k6[255],k7[255],k8[255]
 25: str k9[255],k10[255],s1[255]
 26: str c_u,c_d,c_l,c_r
 27: int kt                    /*ケンタロウ体力
 28: int sc                    /*スコア
 29: int atari                 /*ケンタロウの攻撃が当たった
 30: int jk
 31: int ro
 32: int n
 33: int l                     /*敵キャラクター番号
 34: int x                     /*ケンタロウX座標
 35: int tx                    /*敵のX座標
 36: int ts                    /*敵の体力
 37: int rt                    /*シンケッシュウを突かれた？
 38: int wk                    /*ケンタロウが選択している技
 39: int i
 40: int hn                    /*ケンタロウの死に方
 41: int juf=0                 /*前ジャンプフラグ
 42: int xt                    /*タイム計測用
 43: int at
 44: int q1,q2                 /*乱数用ワーク
 45: int nb
 46: int ne
 47: int nj
 48: int i1                    /*ボーナス計算用ワーク
 49: int bb                    /*ボーナスポイント
 50: int a
 51: int z                     /*技が当たったかどうかのフラグ
 52: int v
 53: int r4
 54: int jt=0
 55: int bn
 56: int t_death=0             /*敵が死んだ
 57: int death=0               /*ケンタロウが死んだ
 58: int tm_cancel             /*敵の移動キャンセル
 59: int uy=0                  /*ハートの攻撃頻度
 60: int ya=0                   /*矢の発射フラグ
 61: int yax=0                  /*矢の表示座標
 62: int over                   /*ゲームオーバー
 63: str dm_key
 64: /*
 65: /* メインループ
 66: /*
 67: int rep
 68:    rep=0
 69:    width 64
 70:    while rep=0
 71:      repeat
 72:         opening()  /*オープニング表示
 73:       . a=sousa()  /*操作説明表示
 74:      until a=0
 75:      data_init()   /*データ初期化
 76:      l=0           /*最初の敵はザコだよ
 77:      kt=5000       /*ケンタロウの初期体力
 78:      over=0
 79:      sc=0
 80:      while over=0
 81:        main_rtn()
 82:      endwhile
 83:    endwhile
 84: end
 85: /*
 86: /* メイン処理
 87: /*
 88: func main_rtn()
 89: int q1,q2,a=0
 90:    init_back()                       /*各種変数&背景初期化
 91:    para_put()                        /*パラメータ表示
 92:    while death=0 and t_death=0 /*ケンタロウ死亡OR敵死亡までループ
 93:      para_put2()                      /*パラメータ表示2
 94:      if l=1 then hn=1
 95: /*
 96:       atari=0
 97:       kenta_main()                    /*ケンタロウメインルーチン
 98:       if atari<>0 then {              /*攻撃が当たったときのスコア加算
 99:          sc=sc+sw(atari-1)
100:          q1=int(rnd()*100) mod 10 + 1 /*1/10で体力増加
101:          q2=int(rnd()*10 )+1
102:          if q1=1 then kt=kt+50*q2
103:          para_put()
104:       }
105:       if kt<=0 then death=100    /*kt(ケンタロウの体力)=0なら死亡
106:       if juf=0 then {
107:         locate x,15
108:         print k(n)           /*ケンタロウ表示
109:       }
110:                              /*敵とケンタロウが重なっているか?
111:       if tx-1=x and death=0 and t_death=0 then {
112:          kt=kt-10*l          /*ケンタロウの体力マイナス
113:          /*music "-#d-#c"
114:          kenta_down()        /*ケンタロウ倒れかけアニメーション
115:       }
116:       if death=0 then {
117:         enemy_main()            /*敵メインルーチン
118:          if t_death<>0 then {
119:             enemy_death_call() /*敵死んだ
120:          }
121:       }
122: /*
```

```
123: /*シンケッシュウを突かれて3分たったら死亡
124:         if (rt=1) and (xt>3600) then death=100
125:         if ya<>0 then ya_move()     /*矢の移動
126:         if kt<=0 then death=100     /*kt(ケンタロウの体力)=0なら死亡
127:     for i=0 to 20:dm_key=inkey$(0):next
128:     if comp=1 then wait(5) else wait(0)
129:     endwhile
130:     if t_death<>0 then stage_clear() /*敵が死んでいたらステージクリア
131:     if death<>0 then kenta_death()   /*ケンタロウが死んだ
132: endfunc
133: /*
134: /* ステージクリア
135: /*
136: func stage_clear()
137: int a
138:     kt=kt+l*100
139:     locate x,15:print k(1)
140:     a=int(rnd()*10) mod 4
141:     locate x-3,12:print win(a):wait(20)
142:     locate 15,10:print "STAGE CLEAR"
143:     wait(120)
144:     if l=10 then {         /*すべての敵を倒した
145:        congra()
146:     }
147: endfunc
148: /*
149: /* おしまい?
150: /*
151: func congra()
152:        locate 5,10
153:        print "ｹﾝﾀﾛｳ ﾊ ｻｲｷｮｳ ﾉ ﾃｷ ﾗｵｳ ｦ ﾀｵｼﾀ｡":wait(120)
154:        locate 5,10
155:        print "ｼｶｼ ｼﾝ ﾉ ｼｭｷﾞｮｳ ﾊ ｺﾚｶﾗ ﾊｼﾞﾏﾙ｡ ":wait(120)
156:        locate 5,10
157:        print "ｻﾗﾅﾙ ﾃｷ ｦ ﾓﾄﾒﾃ ｶﾚ ﾊ ﾀﾋﾞﾀﾞｯﾀ.....":wait(200)
158:        l=0
159: endfunc
160: /*
161: /* ケンタロウ死亡
162: /*
163: func kenta_death()
164:     kt=0:para_put()
165:     switch hn
166:        case 1:kd_1():break
167:        case 2:kd_2():break
168:        case 3:kd_3():break
169:     endswitch
170:     game_over()
171: endfunc
172: /*
173: /* game over
174: /*
175: func game_over()
176: str k
177:      locate 14,3:print "GAME OVER"
178: /*  music "f6er2e6dr3f8g#e6e6d9r0"
179:      locate 11,5:print "YOUR SCORE =";sc*10
180:      k=""
181:      i=0
182:      locate 11,8:print "PUSH [R] KEY !!"
183:      rep=200
184:      while k="" and i<600
185:        k=inkey$(0)
186:        i=i+1
187:        locate 24,9:print ".";c_l;"."
188:        v_disp()
189:      endwhile
190:      if k="r" or k="R" or i=600 then rep=0
191:      over=100
192: endfunc
193: /*
194: /* ザコにやられた
195: /*
196: func kd_1()
197:   locate x,14
198:   print "    ";c_d;c_l;c_l;c_l;c_l;"    ";
199:   print c_d;c_l;c_l;c_l;c_l;"¥ ｢ ";
200:   print c_d;c_l;c_l;c_l;c_l;"-HH*";
201:   locate tx,14:print s(1)
202:   wait(30)    /*music "-c-c"
203:   for i=1 to 4
204:     locate tx-2,12:print zs(i):wait(20)
205:   next
206:   wait(20)
207: endfunc
208: /*
209: /* ホクトシンケンでやられた
210: /*
211: func kd_2()
212:   locate tx,14:print s(1)
213:   locate  x,15:print k(1)
214:   locate x,12:print "ﾎﾞｺｯ":wait(10)
215:   locate x,12:print "    "
216:   locate x+2,13:print "!･"
217:   wait(30)
218:   locate x+1,14:print " @";c_d;c_l;c_l;"/¥";c_d;c_l;c_l;
219:   locate x+1,14:print "¥/";c_d;c_l;c_l;"/|"
220:   wait(60):beep
221:   locate x-1,13:print"¥ﾕﾘｱｧ-../"
222:   locate x+1,12
223:   print "    ";c_d;c_d;c_l;c_l;c_l;c_l;"｢･.｣";
224:   print c_d;c_l;c_l;c_l;c_l;".:'.";
225:   print c_d;c_l;c_l;c_l;c_l;".|h｣";
226:   print c_d;c_l;c_l;c_l;c_l;"｡|¥,";
227:   wait(20) /*music"-f-g"
228:   if l=10 then {
229:      locate tx-1,12:print "ｵﾏｴ ﾊ":wait(30)
230:      locate tx-1,12:print "ｵﾚﾆ ﾊ":wait(30)
231:      locate tx-1,12:print "ｶﾃﾝ!!":wait(55)
232:   }
233: endfunc
234: /*
235: /* ナントセイケンでやられた
236: /*
237: func kd_3()
238:    locate x-3,12:print " ｽﾊﾟｯ.ｽﾊﾟｯ"
239:    locate x+1,14:print "¥･";c_d;c_l;c_l;c_l;"/E-"
240:    locate x+1,16:print "v/";c_d;c_l;c_l;c_l;" ¥ ":wait(20)
241: endfunc
242: /*
243: /* 敵メインルーチン
244: /*
245: func enemy_main()
246: int zg,s1,s2,ff,wr
247:    tm_cancel=0
248:    if l=1 then hn=1
249:    if t_death=0 then {    /*敵は生存している?
250:     if tx<3 then tx=3     /*敵の移動範囲チェック
251:     locate tx,14
252:     print s(1)            /*敵キャラクターの表示
253:     switch l              /*それぞれの敵攻撃ルーチンへ
254:       case 2:spd_attack():break    /*スペードの攻撃ルーチン
255:       case 3:daiya_attack():break  /*ダイヤの攻撃ルーチン
256:       case 4:club_attack():break   /*クラブの攻撃ルーチン
257:       case 5:hart_attack():break   /*ハートの攻撃ルーチン
258:       case 6:ray_attack():break    /*レイの攻撃ルーチン
259:       case 7:jyagi_attack():break  /*ジャギの攻撃ルーチン
260:       case 8:yuda_attack():break   /*ユダの攻撃ルーチン
261:       case 9:shu_attack():break    /*シュウの攻撃ルーチン
262:       case 10:raou_attack():break  /*ラオウの攻撃ルーチン
263:    endswitch
264:    if tx<3 then tx=3           /*敵の移動範囲チェック
265:    if tm_cancel=0 then {       /*敵の移動キャンセル?
266:      if (x-tx)<0 then zg=2 else zg=-2
267:      wr=zg+6:s1=int(rnd()*100) mod wr +1
268:      wr=-zg+6:s2=int(rnd()*10) mod wr +1
269: /*   if my<>1 then {
270: /*     ts=300*(ne+1):my=1
271: /*   }
272:      if ts<=0 then {      /*敵体力<=0?
273:         t_death=1:locate 10,5:print "aaa"
274:      } else {
275:         ff=0
276:         if (s1<=3) and (tx<33) then tx=tx+1:ff=10
277:         if (s2<=3) and (tx>0)  then tx=tx-1:ff=10
278:         if s2>=5 and juf=0 and ff=0 then enemy_seiken()
279:      }
280:    }
281:   }
282: endfunc
283: /*
284: /* 敵のセイケンヅキ
285: /*
286: func enemy_seiken()
287:      locate tx+2,15:print "--]":ts=ts-1:wait(5)
288:      if (x-tx>0) and (x-tx<4) then {  /*ケンタロウに当たっているか?
289:         kt=kt-20*l
290:         kenta_down()            /*ケンタロウダウン?
291:      }
292:      locate tx+2,15:print "   ":ts=ts-1
293: endfunc
294: /*
295: /* スペードの攻撃ルーチン
296: /*
297: func spd_attack()
298: int r,r1
299:   hn=1
300:   if ya=0 and juf=0 then {       /*矢がすでに発射されているか?
301:      r1=int(rnd()*10) + 1
302:      if r1=1 then {
303:         beep
304:         locate tx+1,14:print ">-O>"
305:         wait(20)
306:         yax=tx+1
307:         ya=100
308:         tm_cancel=10
309:      } else {
310:        if r1<3 then tm_cancel=10
311:      }
312:   }
313: endfunc
314: /*
315: /* 矢の移動ルーチン
316: /*
317: func ya_move()
318:   locate yax,14:print "    "
319:   if t_death=0 then {
320:     if yax>=36 then {
321:      yax=0:ya=0
322:     } else {
323:      locate tx,14:print s(1)
324:      if juf=0 then {
325:        locate x,15:print k(1)
326:      }
327:      if comp=1 then yax=yax+2 else yax=yax+4
328:      locate yax,14:print ">-O>"
329:      if (x-yax <= 2) and (x-yax >= -1) then {
330:         kt=0   /*矢が健太郎に当たった
331:      }
332:     }
333:   } else {
334:      yax=0:ya=0
335:   }
336: endfunc
337: /*
338: /* ダイヤの攻撃ルーチン
339: /*
340: func daiya_attack()
341: int r8,r9,bon
342:   r8=int(rnd()*10) mod 4
343:   hn=1
344:   if r8<>0 then bon=r8
345:   if r8=1 and juf=0 and x-tx<8 then {
346:     r9=int(rnd()*10)+1
347:     if r9=1 then {
348:        locate tx-2,12:print "ｺﾉﾎﾞｳｶﾞ":wait(20)
349:        locate tx-2,12:print "ﾐｷﾚﾙｶ!!":wait(25)
350:        locate tx-2,12:print "       "
351:     }
352:     locate tx+2,14:print bo(bon):wait(20)
353:     if x-tx<7 and x-tx>0 then {
354:       locate x,12:print "ﾍﾞｷ"
355:       kt=kt-450:wait(25) /*music "-c-d-d"
356:       kenta_down()
357:     }
358:       locate tx+2,14:print bk(bon)
359:       locate x,12:print "   "
360:   }
361:   r8=int(rnd()*10) mod 6
362:   if r8<4 then tm_cancel=1
363: endfunc
364: /*
365: /* クラブの攻撃ルーチン
366: /*
367: func club_attack()
368: int km,p,u
369:   hn=1
370:   tm_cancel=0
371:   locate  x,15:print k(1)
372:   locate tx,14:print s(1)
373:   km=int(rnd()*10) mod 2
374:   if (km=1) and (juf=0) then {
375:      p=int(rnd()*100) mod 15 + 1
376:      if tx+p>37 then p=0
377:      for i=1 to p
378:        locate  x,15:print k(1)
379:        locate tx+i,14:print s(1)
380:      next
381:      tx=tx+p
382:      locate tx+2,14:print "  "
383:      locate tx+2,15:print "/¥":wait(2)
384:      locate tx+2,15:print "  "
385:      locate tx+2,16:print "/¥":wait(2)
386:      locate tx+2,16:print "  "
387:      if x-tx>=0 and x-tx<=2 then {  /*当たり?
388:        locate x,12:print "ｸﾞｻ":beep
```

▶NEO・GEOを買いました。若いギースにも会いました。でもやっぱり，X68000で「龍虎の拳」がやりたかったよー。　張　徳川(26)千葉県

```
389:         for i=3 to 20
390:           wait(0)           /*画面フラッシュ
391:         next
392:         kt=kt-500
393:         locate x,12:print "   "
394:         kenta_down()
395:       }
396:       for i=1 to p
397:         locate  x,15:print k(1)
398:         locate tx-i,14:print s(1)
399:       next
400:       tx=tx-p
401:       u=int(rnd()*10) mod 6 + 1
402:       if u<3 then tm_cancel=1
403:   }
404: endfunc
405: /*
406: /* ハートの攻撃ルーチン
407: /*
408: func hart_attack()
409: int r5
410:   hn=1
411:   locate tx,14:print s(1)
412:   r5=int(rnd()*100) mod (10-uy) + 1
413:   uy=uy+1
414:   if uy>=8 then uy=8
415:   if r5=1 and juf=0 then {
416:       r5=int(rnd()*10) mod 3 + 1
417:       locate tx+2,14:print "/";c_u;"X"
418:       if r5=1 then {
419:         locate tx-1,12:print "ｲﾃｪﾖｫ"
420:       }
421:       wait(2)
422:       locate tx+2,14:print " ";c_u;" "
423:       locate tx+3,15:print "--+":wait(2)
424:       locate tx-1,12:print "     "
425:       if x-tx>=1 and x-tx<=2 then {
426:          kt=kt-850 /*music."-#e#g"
427:          kenta_down()
428:       }
429:       locate tx+3,15:print "   "
430:   }
431: endfunc
432: /*
433: /* レイの攻撃ルーチン
434: /*
435: func ray_attack()
436: int y2
437:   hn=3
438:   y2=int(rnd()*10) mod 3 + 1
439:   if juf=0 and y2<>1 then {
440:       for i=0 to 3
441:         locate tx+2,14:print "///":wait(0)
442:         locate tx+2,14:print "｢｢｢":wait(0)
443:         locate tx+2,14:print "   "
444:         locate x,15:print k(1)
445:         locate tx+2,15:print "｣｣｣":wait(0)
446:         locate tx+2,15:print "¥¥¥":wait(0)
447:         locate tx+2,15:print "   "
448:         locate x,15:print k(1)
449:       next
450:       if (x-tx>0) and (x-tx<7) then {
451:         locate tx,14:print s(1)
452:         locate x,15:print k(n)
453:         locate tx,12:print "ﾅﾝﾄ":wait(20)
454:         locate tx-2,12:print "ｽｲﾁｮｳｹﾝ":wait(30)
455:         locate tx-2,12:print "       "
456:         kt=kt-1500
457:         kenta_down()
458:         tm_cancel=10
459:       } else {
460:         y2=int(rnd()*10) mod 6 + 1
461:         if y2<3 then tm_cancel=10
462:       }
463:   }
464: endfunc
465: /*
466: /* ジャギの攻撃ルーチン
467: /*
468: func jyagi_attack()
469: int y2
470:   hn=2
471:   if juf=0 then{
472:      y2=int(rnd()*10) mod 7 + 1
473:      if y2=1 then {
474:         if (x-tx>=3) and (x-tx<=5) and tx<27 then {
475:            jyagi_rakan()     /*ホクトラカンゲキ
476:         } else {
477:            if (x-tx>=2) and (x-tx<=4) then {
478:                jyagi_nanto() /*ナントセイケン
479:            }
480:         }
481:      }
482:   }
483: endfunc
484: /*
485: /* ジャギのホクトラカンゲキ
486: /*
487: func jyagi_rakan()
488:   tx=tx+1
489:   locate tx,14:print s(1)
490:   locate tx,12:print "ﾊｧｧ!!"
491:   for i=1 to 6
492:     locate tx+2,14:print "/¥":wait(2)
493:     locate tx+2,14:print "  "
494:     locate  x,15:print k(1)
495:     locate tx+2,15:print "--":wait(2)
496:     locate tx+2,15:print "  "
497:     locate  x,15:print k(1)
498:     locate tx+2,16:print "¥/":wait(2)
499:     locate tx+2,16:print "  "
500:     locate  x,15:print k(1)
501:   next
502:   locate tx,12:print "     "
503:   locate tx,14:print "   ";c_d;c_l;c_l;c_l;"   ";
504:   print c_d;c_l;c_l;c_l;"   ";
505:   print c_d;c_l;c_l;c_l;"   ";
506:   locate  x,15:print k(1)
507:   tx=tx+7
508:   locate tx,14:print s(1)
509:   locate tx,12:print " ﾎｸﾄ ":wait(20)
510:   locate tx,12:print "ﾗｶﾝｹﾞｷ":wait(56)
511:   locate tx,12:print "      "
512:   kt=kt-900
513:   locate x,12:print "ｹﾞﾌ!!":wait(40)
514:   locate x,12:print "      "
515:   tm_cancel=10
516: endfunc
517: /*
518: /* ジャギのナントセイケン
519: /*
520: func jyagi_nanto()
521:   locate tx,14:print s(1)
522:   locate  x,17:print "  "
523:   locate  x,14:print k(n)
524:   tx=tx+1
525:   locate tx,14:print s(1)
526:   locate tx+1,12:print "ﾊｯｯ!":wait(10)
527:   locate tx+2,15:print "---"
528:   locate x+2,15:print "H"
529:   wait(30)
530:   hn=3
531:   kt=kt-400
532:   locate tx+1,12:print "    "
533:   locate x,12:print "ｺﾞﾌｯ!":wait(20)
534:   locate tx+2,15:print " H "
535:   locate x,15:print k(n)
536:   locate tx,14:print s(1)
537:   locate x,12:print "      "
538:   locate x,13:print "   "
539:   x=x-1
540:   tm_cancel=10
541: endfunc
542: /*
543: /* ユダの攻撃ルーチン
544: /*
545: func yuda_attack()
546: int y2,y3
547:   hn=3
548:   if juf=0 then {
549:      y2=int(rnd()*10) mod 7 + 1
550:      if y2=1 then {
551:         y3=int(rnd()*10) mod 2 + 1
552:         if tx<7 then {
553:            if y3<>1 then yuda_for()
554:         } else {
555:            if y3<>1 then yuda_back()
556:         }
557:      } else {
558:        if tx>=23 then {
559:           yuda_back()
560:        } else {
561:           if y2=5 then {
562:              yuda_koukaku()
563:           }
564:        }
565:      }
566:   }
567: endfunc
568: /*
569: /* ユダナントコウカクケン
570: /*
571: func yuda_koukaku()
572: int sq
573:   sq=int(rnd()*100) mod 15 + 1
574:   if x>34 then sq=35-tx
575:   for i=0 to 1
576:      locate tx+2,15:print "｢｢":wait(0) /*music"+#c"
577:      locate tx+2,15:print "｣｣":wait(0) /*music"+#c"
578:      locate tx+2,15:print "¥¥":wait(0) /*music"+#c"
579:      locate tx+2,15:print "//":wait(0) /*music"+#c"
580:   next
581:   for i=0 to sq
582:     locate tx+3+i,17:print "W"
583:   next
584:   if (tx+sq+2>=x) and (tx<x) then {
585:      locate x,12:print "ｳｯ!":wait(10)
586:      locate x,12:print "   "
587:      kt=kt-650
588:      locate tx,12:print "ﾅﾝﾄ":wait(20)
589:      locate tx-2,12:print "ｺｳｶｸｹﾝ ":wait(40)
590:      locate tx-2,12:print "       "
591:   }
592:   for i=0 to sq
593:     locate tx+3+i,17:print
594:   next
595:   tm_cancel=10
596: endfunc
597: /* ユダ前進
598: func yuda_for()
599:    for i=0 to 2
600:       locate tx+i,14:print s(1)
601:    next
602:    tx=tx+2
603: endfunc
604: /* ユダ後退
605: func yuda_back()
606:    for i=0 to 2
607:       locate tx-i,14:print s(1)
608:    next
609:    tx=tx-2
610: endfunc
611: /*
612: /* シュウの攻撃ルーチン
613: /*
614: func shu_attack()
615: int y3,zx
616:   hn=3
617:   if juf=0 then{
618:      y3=int(rnd()*10) mod 7 + 1
619:      if y3>1 then {
620:         zx=x-tx
621:         switch zx
622:          case 7:shu_kyaku():break /*ナントセイケンのキャク
623:          case 5:shu_yugen():break /*ナントユウゲンショウ
624:          case 3:shu_rekya():break /*レッキャククウプ
625:          case 2:shu_rikya():break /*リッキャククウプ
626:         endswitch
627:      }
628:   }
629: endfunc
630: /*
631: /* ナントセイケンのキャク
632: /*
633: func shu_kyaku()
634:   if tx<=18 then {
635:      locate tx,15:print k7
636:      locate tx+2,13:print " 0";c_d;c_l;"Hh";c_d;c_l;c_l;"H";c_d;
c_l;c_l;
637:                     print "//";c_d;c_l;c_l;c_l;"/":wait(10)
638:      locate tx+2,13:print "  ";c_d;c_l;"  ";c_d;c_l;c_l;" ";c_d;
c_l;c_l;
639:                     print "  ";c_d;c_l;c_l;c_l;" "
640:      locate tx+4,10:print "ﾋｮｳ!"
641:      locate tx+4,12:print " 0";c_d;c_l;c_l;"¥H";c_d;c_l;"H--/";
642:                     print c_d;c_l;c_l;c_l;c_l;"<":wait(10)
643:      locate tx+4,12:print "  ";c_d;c_l;c_l;"  ";c_d;c_l;"    ";
644:                     print c_d;c_l;c_l;c_l;c_l;" "
645:      locate tx+4,12:print " 0";c_d;c_l;c_l;" Hh";c_d;c_l;c_l;c_l
;c_l;c_l;
646:                     print "¥--H";c_d;c_l;">":wait(10)
647:      locate tx+4,12:print "  ";c_d;c_l;c_l;"    ";c_d;c_l;c_l;c_l
;c_l;c_l;
648:                     print "    ";c_d;c_l;" "
649:      locate tx+4,10:print "    "
650:      tx=tx+10:locate tx,14:print s(1)
```

▶CD-ROMドライブでWindows/Macintosh用のいやらCDが見られるなら買おうと思ってしまった……。周りにはPhotoCDということにして。　　秋定　貴文(20)兵庫県

```
651:       kt=kt-500
652:       kenta_down()
653:       tm_cancel=10
654:   }
655: endfunc
656: /*
657: /* ナントユウゲンショウ
658: /*
659: func shu_yugen()
660:   locate tx-2,15:print "｣｣｣H｣｣｣":wait(5)/*music "r"
661:   locate tx-2,14:print " ｣ 0 ｣ ":wait(5)/*music "+#c"
662:   locate tx-2,13:print " ｣｣ ｣｣ ":wait(5)/*music "+#d"
663:   locate tx-2,13:print "         ":wait(5)/*music "+#e"
664:   locate tx-2,14:print "   0   ":wait(5)/*music "+#g"
665:   locate tx-2,15:print "   H   ":wait(5)/*music "+#b"
666:   locate tx,15:print k7
667:   tx=tx+3:kt=kt-1000
668:   locate tx,14:print s(1)
669:   locate tx+2,14:print "==@:."
670:   locate x,12:print "ｸﾞｱ!!":wait(30)
671:   locate tx+2,14:print "  *  "
672:   locate x,12:print "      "
673:   locate tx-1,12:print " ﾅﾝﾄ ":wait(20)
674:   locate tx-2,12:print "ﾕｳｹﾞﾝｼｮｳ":wait(20)
675:   locate tx-2,12:print "          "
676:   kenta_down()
677:   tm_cancel=10
678: endfunc
679: /*
680: /* レッキャククウブ
681: /*
682: func shu_rekya()
683:   locate tx,15:print k7
684:   locate tx,12:print "ﾎﾔｰ!!"
685:   for i=0 to 10
686:     locate tx-1,14:print " /";c_d;c_l;c_l;"0HH___";c_d;c_l;c_l;c
_l;" ";c_d;c_l;" "
687:     locate  x+3,12:print "ｶﾞﾌｯ!":wait(0)
688:     locate tx-1,14:print "  ";c_d;c_l;c_l;"      ";c_d;c_l;c_l;c
_l;" ";c_d;c_l;" "
689:     locate  x+3,12:print "      "
690:     locate tx-1,14:print "   ¥";c_d;c_l;c_l;c_l;c_l;"/_HH0";c_d;
c_l;c_l;c_l;" ";c_d;c_l;" "
691:     wait(0)
692:     locate tx-1,14:print "    ";c_d;c_l;c_l;c_l;c_l;"      ";c_d;
c_l;c_l;c_l;" ";c_d;c_l;" "
693:   next
694:   locate tx,12:print "      "
695:   locate tx,14:print s(1)
696:   locate tx,12:print " ﾅﾝﾄ ":wait(20)
697:   locate tx-1,12:print "ﾚｯｷｬｸ ｸｳﾌﾞ ":wait(20)
698:   locate tx-1,12:print "            "
699:   kt=kt-800
700:   kenta_down()
701:   tm_cancel=10
702: endfunc
703: /*
704: /* リッキャククウブ
705: /*
706: func shu_rikya()
707:   locate tx,12:print "ｾｲｯｯ.!"
708:   for i=0 to 10
709:     locate tx+1,13
710:     print "||";c_d;c_l;c_l;"||";c_d;c_l;c_l;"H";
711:     print c_d;c_l;"H0";c_d;c_l;c_l;c_l;"/| "
712:     wait(0)
713:     locate tx+1,13
714:     print "  ";c_d;c_l;c_l;"  ";c_d;c_l;c_l;" ";
715:     print c_d;c_l;"  ";c_d;c_l;c_l;c_l;"   "
716:     locate tx+1,13
717:     print "| /";c_d;c_l;c_l;c_l;"|/";c_d;c_l;c_l;"H";
718:     print c_d;c_l;"H0";c_d;c_l;c_l;c_l;"/|"
719:     wait(0)
720:     locate tx+1,13
721:     print "   ";c_d;c_l;c_l;c_l;"  ";c_d;c_l;c_l;" ";
722:     print c_d;c_l;"  ";c_d;c_l;c_l;c_l;"  "
723:     locate tx+1,13
724:     print "|";c_d;c_l;"|--";c_d;c_l;c_l;c_l;"H";c_d;c_l;"H0";
725:     print c_d;c_l;c_l;c_l;"/|"
726:     wait(0)
727:     locate tx+1,13
728:     print " ";c_d;c_l;"   ";c_d;c_l;c_l;c_l;" ";c_d;c_l;"  ";
729:     print c_d;c_l;c_l;c_l;"  "
730:     locate tx+1,13
731:     print "|";c_d;c_l;"|";c_d;c_l;"H¥";c_d;c_l;c_l;"H0¥";
732:     print c_d;c_l;c_l;c_l;c_l;"/|"
733:     wait(0)
734:     locate tx+1,13
735:     print " ";c_d;c_l;" ";c_d;c_l;"  ";c_d;c_l;c_l;"   ";
736:     print c_d;c_l;c_l;c_l;c_l;"  "
737:     locate tx+1,13
738:     print "|";c_d;c_l;"|--";c_d;c_l;c_l;c_l;"H";c_d;c_l;"H0 ";
739:     print c_l;c_l;c_l;"/|"
740:     wait(0)
741:     locate tx+1,13
742:     print " ";c_d;c_l;"   ";c_d;c_l;c_l;c_l;" ";c_d;c_l;"
743:     print c_l;c_l;c_l;"  "
744:     locate tx+1,13
745:     print "| /";c_d;c_l;c_l;c_l;"|/";c_d;c_l;c_l;"H";c_d;c_l;"H0
";
746:     print c_d;c_l;c_l;c_l;"/|"
747:     wait(0)
748:     locate tx+1,13
749:     print "   ";c_d;c_l;c_l;c_l;"  ";c_d;c_l;c_l;" ";c_d;c_l;"
";
750:     print c_d;c_l;c_l;c_l;"  "
751:     locate tx+1,13
752:     print "||";c_d;c_l;c_l;"||";c_d;c_l;"H";c_d;c_l;"H0".
753:     print c_d;c_l;c_l;c_l;"/|"
754:     wait(0)
755:     locate tx+1,13
756:     print "  ";c_d;c_l;c_l;"  ";c_d;c_l;" ";c_d;c_l;"  ";
757:     print c_d;c_l;c_l;c_l;"  "
758:   next
759:   locate tx,14:print s(1)
760:   locate tx,12:print "ﾅﾝﾄ   ":wait(10)
761:   locate tx-1,12:print "ﾘｯｷｬｸｸｳﾌﾞ":wait(10)
762:   locate tx-1,12:print "           "
763:   kt=kt-900
764:   kenta_down()
765:   tm_cancel=10
766: endfunc
767: /*
768: /* ラオウの攻撃ルーチン
769: /*
770: func raou_attack()
771: int r
772:   hn=2
773:   if juf=0 then {
774:     r=int(rnd()*10) mod 5 + 1
775:     switch r
776:       case 2:raou_at2()
777:       case 3:raou_at3()
778:       case 4:raou_at1()
779:       case 5:raou_at1()
780:     endswitch
781:   }
782: endfunc
783: /*
784: /* ラオウ攻撃1
785: /*
786: func raou_at1()
787:   if (x-tx=3) or (x-tx=4) then {
788:     for i=0 to 20
789:      locate tx,12:print "ｱﾀ!"
790:      locate tx+2,15:print "--¥":wait(0)
791:      locate tx+2,15:print "--/":wait(0)
792:      locate tx+2,15:print "   ":wait(0)
793:      locate tx,12:print "   "
794:     next
795:     locate x-1,12:print "ﾌﾞｳ!":wait(5)
796:     locate x-1,12:print "     "
797:     kt=kt-1100
798:     tm_cancel=10
799:   }
800: endfunc
801: /*
802: /* ラオウ攻撃2
803: /*
804: func raou_at2()
805:   if x-tx=2 then {
806:     locate tx,15:print k7
807:     locate tx,14:print " o";c_d;c_l;c_l;"¥H";c_d;c_l;"H";c_d;c_
l;c_l;"_|"
808:     wait(5) /*music "rr"
809:     locate tx,14:print "  ";c_d;c_l;c_l;"  ";c_d;c_l;" ";c_d;c_
l;c_l;"  "
810:     locate tx,12:print "ｱﾀｱ!!"
811:     locate tx,14:print " o";c_d;c_l;c_l;"/H｣/";c_d;c_l;c_l;c_l;
"H/";c_d;c_l;c_l;"|"
812:     wait(10) /*music "r-#f-#c-#g"
813:     locate tx,14:print "  ";c_d;c_l;c_l;"    ";c_d;c_l;c_l;c_l;
"  ";c_d;c_l;c_l;" "
814:     locate tx,12:print "      "
815:     locate tx,14:print s(1)
816:     kenta_down()
817:     kt=kt-1300
818:     tm_cancel=10
819:   }
820: endfunc
821: /*
822: /* ラオウ攻撃3
823: /*
824: func raou_at3()
825:   if x-tx=2 and rt=0 then {
826:     locate tx+2,15:print "｣ ';c_u;c_l;"｢¯¯":wait(20)/*music"-e-d
"
827:     locate tx-2,12:print "ﾋｺｳ ﾉ ﾋﾄﾂ":wait(20)
828:     locate tx-2,12:print "ｼﾝｹｯｼｭｳ ｦﾂｲﾀ":wait(25)
829:     locate tx-2,12:print "            "
830:     locate tx+2,15:print " ";c_u;c_l;"   "
831:     locate tx,15:print k7
832:     if tx>4 then tx=tx-3:locate tx,14:print s(1)
833:     rt=1:xt=0            /*  ここで時間初期設定
834:     tm_cancel=10
835:   }
836: endfunc
837: /*
838: /* 敵死んだコール
839: /*
840: func enemy_death_call()
841:   switch wk
842:     case 1:enemy_death():break
843:     case 2:enemy_death():break
844:     case 3:enemy_death2():break
845:     case 4:enemy_death():break
846:     case 5:enemy_death2():break
847:     case 6:enemy_death():break
848:     case 7:enemy_death():break
849:     case 8:enemy_death2():break
850:   endswitch
851: endfunc
852: /*
853: /*
854: /* 敵死んだ1
855: /*
856: func enemy_death()
857:   locate tx,14:print s1:wait(5) /*music "-d"
858:   locate x,15:print k(n)
859:   locate tx,17:print "XHH/¥"
860:   locate tx-1,15:print "ﾄﾞｻｯ"
861:   wait(80)
862:   enemy_death2()
863: endfunc
864: /*
865: /* 敵死んだ2
866: /*
867: func enemy_death2()
868:   tn=tn+1
869:   r2=int(rnd()*10) mod 3
870:   sc=sc+20*r2*1
871:   locate tx,17:print "      "
872:   locate tx-1,15:print "      "
873: endfunc
874: /*
875: /* ケンタロウメインルーチン
876: /*
877: func kenta_main()
878: str kkm
879:   if juf<>0 then {
880:     for_jump_move()
881:   } else {
882:   kkm=inkey$(0)
883:     switch asc(kkm)
884:       case '4':x=x-1:kt=kt-1:break  /*左移動
885:       case '6':x=x+1:kt=kt-1:break  /*右移動
886:       case 'x':wk=wk+1:break        /*技選択
887:       case 'z':wk=wk-1:break        /*技選択
888:       case 'c':kenta_nishi():break  /*ﾆｼ ｼﾝｸｳﾊ
889:       case ' ':kenta_attack():break /*敵にアタック
890:       case '8':jt=1:for_jump_init():break  /*前ジャンプ
891:       case '2':jt=1:back_jump():break      /*後ろジャンプ
892:     endswitch
893:     if wk=9  then wk=1
894:     if wk=0  then wk=8
895:     if x>=36 then x=36
896:     if x<=4  then x=4
897:   }
898: endfunc
899: /*
900: /* ケンタロウニシシンクウハ
901: /*
902: func kenta_nishi()
903:   kt=kt-20
904:   if ya=0 then {
905:     locate x+1,14:print "|";c_d;c_l;"¥":wait(10)
```

▶次期XではポリゴンやテクスチャマッピングがBASICの命令一発でできるといいなぁ。
河野　太郎(21)東京都

```
 906:       locate x+1,14:print " ";c_d;c_l;" "
 907:       kt=kt-3
 908:   } else {
 909:       locate yax,14:print "    "
 910:       locate x,15:print k(1)
 911:       locate x-1,14:print ">-+";c_d;c_l;"¥"
 912:       locate x-2,12:print "ﾎｸﾄｼﾝｹﾝ":wait(20)
 913:       locate x-2,12:print "ﾆｼ ｼﾝｸｳﾊ!":wait(20)
 914:       locate x-2,12:print "          "
 915:       locate x-1,14:print "   ";c_d;c_l;" "
 916:       sc=sc+(int(rnd()*10) mod 5 + 1 )*60
 917:       yax=0;ya=0
 918:   }
 919: endfunc
 920: /*
 921: /* ケンタロウマエジャンプ
 922: /*
 923: func for_jump_init()
 924:   if x>9 then {
 925:       for_jump_move()
 926:   } else {
 927:      jt=0
 928:   }
 929: endfunc
 930: func for_jump_move()
 931: int r
 932:     juf=juf+1
 933:     switch juf
 934:       case 1:jump_first()
 935:       case 2:jump_second()
 936:       case 3:jump_third()
 937:     endswitch
 938: endfunc
 939: /*
 940: /* 初めてジャンプした
 941: /*
 942: func jump_first()
 943:   locate x-1,15:print k1
 944:   locate x-1,17:print "     ";
 945:   locate x-1,12:print k(5):wait(5)
 946: endfunc
 947: /*
 948: /* ジャシプして2回目の動作
 949: /*
 950: func jump_second()
 951:   locate x-1,12:print k2
 952:   locate x-5,10:print k(6):wait(5)
 953: endfunc
 954: /*
 955: /* ジャンプして3回目の動作
 956: /*
 957: func jump_third()
 958: int h
 959:   locate x-5,10:print k3
 960:   locate x-9,14:print k(3)              /*ケリのアニメーション
 961:   h=x-9-tx
 962:   if (h > -4) and (h < 2) then {
 963:      teki_damege()
 964:   } else {
 965:     wait(5)
 966:   }
 967:   locate x-9,14:print k10               /*ケリの消去
 968:   x=x-7
 969:   locate x,15:print k(1)
 970:   juf=0
 971: endfunc
 972: /*
 973: /* 敵が通常攻撃を受けた
 974: /*
 975: func teki_damege()
 976:   beep
 977:   locate tx-2,13:print "ｹﾞｪ!"
 978:   locate tx-1,14:print "¥  "
 979:   locate tx-1,15:print "*¥  "
 980:   locate tx-1,16:print "I¥¥"
 981:   wait(30) /*music "+a+d+a"
 982:   locate tx-2,13:print "    "
 983:   locate tx-1,14:print " "
 984:   locate tx-1,15:print "  "
 985:   locate tx-1,16:print "   "
 986:   locate tx,14:print s(1)
 987: endfunc
 988: /*
 989: /* ケンタロウ後ろジャンプ
 990: /*
 991: func back_jump()
 992:   jt=0
 993:   if x<=27 then {                 /*ジャンプできる範囲
 994:      locate x,15:print k7
 995:      kt=kt-5
 996:      locate tx,14:print s(1)
 997:      x=x+2
 998:      locate x,12:print " @";c_d;c_l;c_l;"/Hh";c_d;c_l;c_l;c_l;"H
";c_d;c_l;"¥¥";c_d;c_l;"¥"
 999:      wait(5)
1000:      locate x,12:print "  ";c_d;c_l;c_l;"   ";c_d;c_l;c_l;c_l;"
";c_d;c_l;"  ";c_d;c_l;" "
1001:      x=x+4
1002:      locate x,10
1003:      print  "/HH¥@";c_d;c_l;c_l;c_l;c_l;c_l;c_l;"//  ¥";c_d;c_l;
c_l;c_l;c_l;c_l;c_l;"/"
1004:      wait(5)
1005:      locate x,10
1006:      print  "     ";c_d;c_l;c_l;c_l;c_l;c_l;c_l;"     ";c_d;c_l;
c_l;c_l;c_l;c_l;c_l;" "
1007:      x=x+2
1008:      locate x,15:print k(n)
1009:   }
1010: endfunc
1011: /*
1012: /* ケンタロウの攻撃
1013: /*
1014: func kenta_attack()
1015:    switch wk
1016:      case 1:kenta_seiken():break /*セイケンヅキ
1017:      case 2:kenta_keri():break   /*ケリ
1018:      case 3:kenta_hikou():break  /*ヒコウを突く
1019:      case 4:kenta_zankai():break /*ホクトザンカイケン
1020:      case 5:kenta_hyaku():break  /*ホクトヒャクレツケン
1021:      case 6:kenta_hagan():break  /*コウシュハガンケン
1022:      case 7:kenta_kaiko():break  /*ホクトカイコツケン
1023:      case 8:kenta_jyuha():break  /*ホクトジュウハザン
1024:    endswitch
1025: endfunc
1026: /*
1027: /* ケンタロウのセイケンヅキ
1028: /*
1029: func kenta_seiken()
1030:   kt=kt-1
1031:   if x-tx>=0 then {        /*向かい合っているときのみ攻撃可能
1032:      locate x-1,15:print "｢--"
1033:      locate x+1,12:print "ｱﾀ!"
1034:      if x-tx<4 then {
1035:         ts=ts-40:beep /*music"c" 敵に当たった
1036:         atari=wk      /*スコア用
1037:         teki_damege()
1038:      } else {
1039:         wait(25)
1040:      }
1041:      locate x-1,15:print "   "
1042:      locate x+1,12:print "   "
1043:    /* ここに乱数で敵の攻撃をキャンセルするルーチンあり
1044:   }
1045: endfunc
1046: /*
1047: /* ケンタロウのケリ
1048: /*
1049: func kenta_keri()
1050:   kt=kt-3
1051:   if x-tx>=0 then {        /*向かい合っているときのみ攻撃可能
1052:     locate x+1,12:print "ｵｳﾀｯ!!"
1053:     kt=kt-1
1054:     locate x-2,16:print k(3) /*ケリのアニメーション
1055:     if x-tx<4 then {
1056:        ts=ts-80:beep /*music "a" 敵に当たった
1057:        atari=wk      /*スコア用
1058:        teki_damege()
1059:     } else {
1060:       wait(25)
1061:     }
1062:     locate x-2,16:print k1
1063:     locate x+1,12:print "      "
1064:   /* ここに乱数で敵の攻撃をキャンセルするルーチンあり
1065:   }
1066: endfunc
1067: /*
1068: /* ヒコウを突く
1069: /*
1070: func kenta_hikou()
1071: int d2,d3
1072:    if x-tx<>2 then {          /*当たり?
1073:       kt=kt-20:z=0
1074:    } else {
1075:       z=1
1076:       atari=wk       /*スコア用
1077:    }
1078:    d2=int(rnd()*10) mod 3 + 1     /*突く位置を決定
1079:    locate x,14
1080:    switch d2
1081:      case 1:print "¥¥";c_d;c_l;c_l;"¥_":break
1082:      case 2:print "--";c_d;c_l;c_l;"__":break
1083:      case 3:print "‾|";c_d;c_l;"|":break
1084:    endswitch
1085:    if z=0 then {
1086:      wait(5)
1087:      locate x,14:print "  ";c_d;c_l;c_l;"  "
1088:    } else {
1089:      locate x-5,12:print w(d2);
1090:      print " ﾄｲｳ ﾋｺｳｦﾂｲﾀ":wait(60)
1091:      locate x-5,12
1092:      print "                "
1093:      kt=kt-10
1094:      if (l=10) or ((l=5) and (d2<>3)) then {
1095:         no_damege()   /*ﾗｵｳ,ﾊｰﾄ(3ﾉﾐ)には効かない
1096:      } else {
1097:         locate x,14:print "  ";c_d;c_l;c_l;"  "
1098:         x=x+2
1099:         locate tx,14:print s(1)
1100:         locate x,14:print " ";c_d;c_l;" ";c_d;c_l;" "
1101:         locate x,15:print k(1)
1102:         locate x,15:print "_-";c_l;c_l;c_l;c_d;c_d;" "
1103:         switch d2
1104:           case 1:ht_m1():break
1105:           case 2:ht_m2():break
1106:           case 3:ht_m3():break
1107:         endswitch
1108:         locate x,12:print "        "
1109:         d3=int(rnd()*10) mod 5 + 1
1110:         switch d3
1111:           case 1:ht_a1():break
1112:           case 2:ht_a2():break
1113:           case 3:ht_a3():break
1114:           case 4:ht_a4():break
1115:           case 5:ht_a5():break
1116:         endswitch
1117:         locate x-2,12:print "              "
1118:         sp_dmg_sub4()  /*爆発!
1119:      }
1120:    }
1121: endfunc
1122: /* ヒコウアクション1
1123: func ht_a1()
1124:   locate tx,12:print "ｿﾝﾅ..":wait(60) /*music"+b+br"
1125:   locate tx,12:print "ﾍﾞﾌﾞﾌﾞ"
1126:   locate tx-1,14:print " @ｒ  ";c_d;c_l;c_l;c_l;c_l;c_l;"｣¥¥  ";
1127:   print c_d;c_l;c_l;c_l;c_l;c_l;" |H ";c_d;c_l;c_l;c_l;c_l;
1128:   print " _ ¥":wait(95)
1129:   locate tx,12:print "      "
1130: endfunc
1131: /* ヒコウアクション2
1132: func ht_a2()
1133:   locate tx,12:print "ﾀｽｹﾃ!":wait(60) /*music"aa"
1134:   locate tx,12:print "      "
1135:   locate x-1,12:print "ｱｸﾄｳ ﾉ":wait(60)
1136:   locate x-1,12:print "ﾋﾒｲ ﾊ ":wait(70)
1137:   locate x-1,12:print "ｷｺｴﾝﾅ!":wait(70)
1138:   locate x-2,12:print "                ":wait(20)
1139:   locate tx-1,12:print "ﾃﾚﾚﾚ.."
1140:   locate tx,14:print "¥    ";c_d;c_l;c_l;c_l;c_l;c_l;"/M¥ ";
1141:   print c_d;c_l;c_l;c_l;c_l;"@VH ";c_d;c_l;c_l;c_l;" //"
1142:   wait(95)
1143:   locate tx-1,12:print "       "
1144: endfunc
1145: /* ヒコウアクション3
1146: func ht_a3()
1147:   locate tx-2,12:print "ｺｲﾂ!!!!":wait(50)
1148:   locate tx-2,12:print "ｯﾗﾗﾗﾗ.."
1149:   locate tx,14:print "L@   ";c_d;c_l;c_l;c_l;c_l;c_l;c_l;";HD ";
1150:   print c_d;c_l;c_l;c_l;c_l;"XX ";c_d;c_l;c_l;c_l;
1151:   print "/ L":wait(95)
1152:   locate tx-2,12:print "        "
1153: endfunc
1154: /* ヒコウアクション4
1155: func ht_a4()
1156:   locate tx-3,12:print "ｲﾀｸﾓ ｶﾕｸﾓﾈｪ":wait(55)
1157:   locate tx-3,12:print " ｺﾚｶﾞ ｱﾉ   ":wait(50)
1158:   locate tx-3,12:print "ﾎｸﾄ ｼﾝｹﾝｶｲ!":wait(58)
1159:   locate tx-3,12:print "ﾜﾗﾜｾﾙｾﾞ.   ":wait(55)
1160:   locate tx-3,12:print "ﾍｯﾍｯｹﾞﾚﾚﾚ  "
1161:   locate tx,14:print ".@   ";c_d;c_l;c_l;c_l;c_l;c_l;",#｣ ";
1162:   print c_d;c_l;c_l;c_l;c_l;" HD";c_d;c_l;c_l;c_l;
1163:   print "// ":wait(95)
1164:   locate tx-3,12:print "          "
1165: endfunc
1166: /* ヒコウアクション5
1167: func ht_a5()
```

▶新宿駅東口の某ゲームセンターに，なんと「空手道」があった。早く出ないかな。
鳩山　智之(18)神奈川県

```
1168:   locate tx-3,12:print "ﾅﾝﾄﾓﾅｲ!":wait(62)
1169:   locate tx-3,12:print "          "
1170:   locate  x-3,12:print "ｷｻﾏﾉﾖｳﾅ ｹﾞﾄﾞｳ":wait(60)
1171:   locate  x-3,12:print "ｵﾚｶﾞｲｶｽﾄ ｵﾓｳｶ":wait(65)
1172:   locate  x-3,12:print "                ":wait(30)
1173:   locate   tx,12:print "ﾋﾍﾞﾍﾞ"
1174:   locate tx,14:print ".¥@  ";c_d;c_l;c_l;c_l;c_l;c_l;".// ";
1175:   print c_d;c_l;c_l;c_l;c_l;"¥¥¥";c_d;c_l;c_l;c_l;
1176:   print "/ ¥":wait(95)
1177:   locate tx-3,12:print "           "
1178:   locate   tx,12:print "      "
1179: endfunc
1180: /* ヒコウメッセージ1
1181: func ht_m1()
1182:   locate x,12:print "ｵﾏｴﾊﾓｳ..":wait(60)
1183:   locate x,12:print "ｼﾝﾃﾞｲﾙ!!":wait(70)
1184: endfunc
1185: /* ヒコウメッセージ2
1186: func ht_m2()
1187:   locate x,12:print "ｵﾏｴﾉｲﾉﾁﾊ":wait(60)
1188:   locate x,12:print "ｱﾄ 5ﾋﾞｮｳ":wait(70)
1189: endfunc
1190: /* ヒコウメッセージ3
1191: func ht_m3()
1192:   locate x,12:print "ｵﾉﾚﾉ ﾂﾐﾌﾞｶｻｦ":wait(60)
1193:   locate x,12:print "ｵﾓｲｼﾚ!!      ":wait(70)
1194: endfunc
1195: /*
1196: /* ホクトザンカイケン
1197: /*
1198: func kenta_zankai()
1199:   kt=kt-40
1200:   if x-tx<>3 then {  /*攻撃が当たったか?
1201:      z=0
1202:   } else {
1203:      z=1
1204:      atari=wk      /*スコア用
1205:   }
1206:   locate x-1,15:print k(n)
1207:   locate x-1,14:print "_-";c_l;c_l;c_d;"|_":wait(10)
1208:   if z<>0 then {
1209:      locate x-1,12:print "ﾎｸﾄｻﾞﾝｶｲｹﾝ":beep /*music "#d#c#d"
1210:      wait(80)
1211:      locate x-2,12:print "3ﾋﾞｮｳｺﾞ ﾆ  ":wait(100)
1212:      locate x-2,12:print " ｵﾏｴﾊ ｼｽ!  ":wait(150)
1213:      r4=int(rnd()*10) mod 7 + 1
1214:      if (l=10) or (l=6) or (l=5) then {
1215:         no_damege()
1216:      } else {
1217:         locate x-2,12:print "              "
1218:         locate x-1,14:print "  "
1219:         locate x-1,15:print "  "
1220:         special_damege()
1221:      }
1222:    } else {
1223:         locate x-2,12:print "              "
1224:         locate x-1,14:print "  "
1225:         locate x-1,15:print "  "
1226:    }
1227: endfunc
1228: /*
1229: /* ホクトヒャクレツケン
1230: /*
1231: func kenta_hyaku()
1232: int fx
1233:   if x-tx<>3 then {
1234:      kt=kt-60:z=0          /*ハズレた
1235:   } else {
1236:      z=1                   /*当たり
1237:      atari=wk      /*スコア用
1238:   }
1239:   locate x-1,15:print k(n)
1240:   locate x-2,12:print "ｱﾀﾀﾀﾀ.."
1241:   for i=0 to 30  /*多分画面フラッシュ処理が入る
1242:     locate x-1,14:print "‾|";c_d;c_l;"|";c_l;c_l;"__"
1243:     wait(0)
1244:     locate x-1,14:print "  ";c_d;c_l;" ";c_l;c_l;"  "
1245:     locate x-1,15:print "--"
1246:     wait(0)
1247:     locate x-1,15:print "  "
1248:     locate x-1,16:print "_｣";c_d;c_l;"ｒ";c_l;c_l;"‾‾"
1249:     wait(0)
1250:     locate x-1,16:print "  ";c_d;c_l;" ";c_l;c_l;"  "
1251:   next
1252:   locate x-2,12:print "        "
1253:   if z<>0 then {            /*当たったときのみ次の処理を行う
1254:     if (l=5) or (l=9) or (l=10) then {  /* ハート, シュウ, ラオウには
ヒャクレツが効かない!
1255:        no_damege()  /*技が効かなかった
1256:     } else {
1257:        locate x-2,12:print "ﾀﾀﾀﾀﾀﾀ."
1258:        locate  tx,13:print s(1)
1259:        locate x-1,15:print k(n)
1260:        locate  tx,17:print "   "
1261:        for i=1 to 110    /*ここでも画面フラッシュ?
1262:           locate x,13:print "¥";c_d;c_l;" ";c_d;c_l;"¥"
1263:           wait(1)
1264:           locate x,13:print " ";c_d;c_l;"¥";c_d;c_l;" "
1265:           wait(1)
1266:        next
1267:        locate x,14:print " "
1268:        locate x-2,12:print "        "
1269:        locate tx-1,13:print s1
1270:        locate tx-2,13
1271:        print "/#--";c_u;c_l;c_l;c_l;c_l;"* /";c_d;c_d;c_l;c_l;"¥
"
1272:        wait(20)
1273:        locate tx-2,13
1274:        print "    ";c_u;c_l;c_l;c_l;c_l;"   ";c_d;c_d;c_l;c_l;"
"
1275:        locate tx-2,17:print "*HH/¥":beep /*music "-d"
1276:        locate tx-2,15:print "ﾄﾞｻｯ"
1277:        locate x,12:print "ﾎｸﾄ"
1278:        wait(80)
1279:        locate x-2,12:print "ﾋｬｸﾚﾂｹﾝ"
1280:        wait(100)
1281:        locate tx-2,17:print "     "
1282:        locate tx-2,15:print "    "
1283:        locate x,12:print "    "
1284:        locate x-2,12:print "        "
1285:        locate tx-1,14:print s(1)
1286:        fx=int(rnd()*10) mod 6 + 1
1287:        switch fx
1288:          case 1:locate tx-3,12:print "ｵﾏｴﾉ ｹﾝ ﾅﾄﾞ":wait(60):loca
te tx-3,12:print "ｶﾎﾄﾞﾓ ｷｶﾝﾊ!":wait(90):break
1289:          case 2:locate tx-3,12:print "ﾍｯﾍｯﾍｯ..":wait(60):locate
tx-3,12:print "ﾅﾝﾄﾓﾅｲｾﾞ!":wait(90):break
1290:          case 3:locate   tx,12:print "ｷｻﾏ!":wait(90):break
1291:          case 4:locate tx-2,12:print "ｺﾚﾊ ｱﾉｯ":wait(60):locate t
x-5,12:print "ﾃﾞﾝｾﾂﾉ ｹﾝﾎﾟｳ":wait(90):locate tx-5,12:print "  ﾎｸﾄｼﾝｹﾝ!  "
:wait(15):break
1292:          case 5:locate tx-2,12:print "ｲﾃｪﾅ ｺﾉ!":wait(90):break
1293:          case 6:locate tx-2,12:print "ｵﾈｶﾞｲﾀﾞ!!":wait(60):locate
tx-2,12:print " ﾀｽｹﾃｸﾚ! ":wait(90):break
1294:        endswitch
1295:        kt=kt-80
1296:        special_damege()  /*必殺技が決まった!
1297:     }
1298:   }
1299: endfunc
1300: /*
1301: /* コウシュハガンケン
1302: /*
1303: func kenta_hagan()
1304: int at=0
1305:   kt=kt-50
1306:   if x>=8 then {          /*技の有効範囲
1307:      if x-tx=2 then at=1 /*当たり?
1308:      locate   x,12:print "ｳｱﾀ"
1309:      locate x-1,15:print "¥__":wait(20)
1310:      locate  tx,14:print s(1)
1311:      locate x-1,17:print "     "
1312:      locate   x,12:print "   "
1313:      locate x-1,15:print "   "
1314:      locate   x,15:print k1
1315:      locate  tx,14:print s(1)
1316:      locate   x,11:print k(4):wait(20)
1317:      locate  tx,14:print s(1)
1318:      locate   x,11:print k4
1319:      locate x-3,10:print k(6):wait(20)
1320:      locate x-3,10:print k6
1321:      locate tx,14:print s(1)
1322:      x=x-6:locate x,15:print k(1)
1323:   }
1324:   if at<>0 then {
1325:      atari=wk      /*スコア用
1326:      at=0                   /*当たったときの処理
1327:      locate x-2,12:print "ｺｳｼｭﾊｶﾞﾝｹﾝ":wait(60)
1328:      locate x-2,12:print "          "
1329:      if (l=5) or (l=6) or (l=9) or (l=10) or (l=7) then {
1330:         no_damege() /*ﾊｰﾄ,ﾚｲ,ｼｭｳ,ﾗｵｳ,ｼﾞｬｷには効かない
1331:      } else {
1332:         locate tx-1,12:print "ｸﾞｷｯ!":beep:wait(20) /*music "-#c-
f"
1333:         locate tx-2,12:print "ｺﾞｷｺﾞｷ.":beep:wait(20) /*music "-
#c0r-#gr-dr-#drr"
1334:         locate tx-3,12:print "
1335:         special_damege()
1336:      }
1337:   }
1338: endfunc
1339: /*
1340: /* ホクトカイコツケン
1341: /*
1342: func kenta_kaiko()
1343: int at
1344:   kt=kt-45
1345:   if x-tx<>2 then at=0 else at=1
1346:   locate x,14:print "‾|";c_d;c_l;"¥":wait(20)
1347:   if at=0 then {
1348:      locate x,14:print "  ";c_d;c_l;" "
1349:   } else {
1350:      atari=wk      /*スコア用
1351:      wait(20) /*music "-#de"
1352:      locate x,12:print "ﾎｸﾄ":wait(55)
1353:      locate x-2,12:print "ｶｲｺﾂｹﾝ":wait(75)
1354:      if (l>4) and (l<10) then {
1355:         locate x-2,12:print "       "
1356:         no_damege()    /*ﾊｰﾄ,ﾚｲ,ｼﾞｬｷﾞ,ﾕﾀﾞには効かない
1357:      } else {
1358:         locate x-2,12:print "       "
1359:         locate x,14:print "  ";c_d;c_l;" "
1360:         special_damege()
1361:      }
1362:   }
1363: endfunc
1364: /*
1365: /* ホクトジュウハザン
1366: /*
1367: func kenta_jyuha()
1368: int g
1369:   locate x,15:print k(1)
1370:   locate x,17:print "   "
1371:   locate tx,14:print s(1)
1372:   locate  x-1,15:print k(7)
1373:   locate  x-1,12:print "ｵﾜﾀﾀﾀﾀ."
1374:   for i=0 to 20
1375:     locate x-1,16:print "___"
1376:     locate x-1,16:print "==="
1377:     locate x-1,16:print "___"
1378:     locate x-1,16:print "---"
1379:   next
1380:   kt=kt-70
1381:   locate x-1,12:print "        "
1382:   locate x-1,16:print "   "
1383:   locate x-1,15:print "   "
1384:   locate   x,13:print "   "
1385:   if x-tx=3 then {
1386:      if (l=6) or (l=8) or (l=10) then {
1387:         no_damege()
1388:      } else {
1389:         locate x-1,16:print ju
1390:         locate x,15:print k(1)
1391:         if l=5 then {       /*ハートがジュウハザンを受けた
1392:            locate x-1,16:print ju
1393:            locate x,15:print k(1)
1394:            locate tx,14:print s(5)
1395:            locate tx,12:print "ｵｵ.":wait(10)
1396:            locate x+1,15:print " "
1397:            locate x-1,15:print "---"
1398:            locate tx,12:print "    "
1399:            locate  x,12:print "ｳｱﾀ!":wait(10)
1400:            locate  x,12:print "    "
1401:            locate x-1,15:print "    "
1402:            locate tx+1,16:print "*"
1403:            locate x-1,12:print "ﾎｸﾄｼﾞｭｳﾊｻﾞﾝ":wait(10)
1404:            g=int(rnd()*10) mod 3 + 1
1405:            locate x-1,12:print "            "
1406:            locate tx-1,12:print hh(g):wait(10)
1407:            locate tx-1,12:print "        "
1408:            kt=kt-15
1409:         }
1410:         atari=wk
1411:         locate x-1,12:print " '     "
1412:         locate x-1,16:print ju
1413:         locate x,15:print k(1)
1414:         locate tx,14:print s(5)
1415:         special_damege()
1416:         tm_cancel=100
1417:      }
1418:   }
1419: endfunc
1420: /*
1421: /* 敵に必殺技が決まったとき
1422: /*
1423: func special_damege()
```

▶Oh!X LIVE inを毎月4曲ぐらいにしてほしい。　大里　博行(20)神奈川県

```
1424:    locate tx,14:print s(1)
1425:    h=0:locate x,15:print k(1)
1426:    v=int(rnd()*20) mod 11 + 1
1427:    switch wk        /*技によって敵の死に方を変える
1428:      case 4:z=0:sp_dmg_sub1():break     /*ホクトザンカイケン
1429:      case 5:z=0:sp_dmg_sub3():break     /*ヒャクレツケン
1430:      case 6:sp_dmg_sub2():break         /*コウシュハガンケン
1431:      case 7:sp_dmg_sub2():break         /*ホクトカイコツケン
1432:      case 8:sp_dmg_sub4():break         /*ホクトジュウハザン
1433:    endswitch
1434:    t_death=1
1435: endfunc
1436: /*
1437: /* 必殺技が決まったときのサブルーチン
1438: /*
1439: func sp_dmg_sub1()
1440:     for i=0 to 2  /*カウントダウン
1441:       locate tx+1,11:print 3-i:beep /* music "+crrr"
1442:       wait(60)
1443:     next
1444:     sp_dmg_sub2()
1445: endfunc
1446: func sp_dmg_sub2()       /*コウシュハガンケン，ホクトカイコツケンの場合
1447:     locate tx-1,12:print sh(v)
1448:     locate tx,13:print ". "
1449:     locate tx,14:print "｢v/":beep  /*music "-e-d"
1450:     wait(150)
1451:     locate tx+2,11:print "  "
1452:     locate tx-1,12:print "
1453:     locate tx,13:print "   "
1454:     locate tx,14:print "   "
1455:     locate tx,14:print s1
1456:     t_death=1
1457: endfunc
1458: func sp_dmg_sub3()      /*ヒャクレツケンの場合
1459:     locate tx-3,12:print "              "
1460:     locate x-2,12:print "ｵﾏｴﾊ ﾓｳ..":wait(80)
1461:     locate x-2,12:print " ｼﾝﾃﾞｲﾙ! ":wait(80)
1462:     locate x-2,12:print "              "
1463:     sp_dmg_sub4()
1464: endfunc
1465: func sp_dmg_sub4()      /*ホクトジュウハザン＆ヒコウの場合
1466: int r5
1467:     r5=int(rnd()*20) mod 11 + 1
1468:     locate tx-1,12:print sh(r5)
1469:     locate tx-1,14:print "｢.･. "
1470:     locate tx-1,15:print "･･h｡ "
1471:     locate tx-1,16:print ". /｣ "
1472:     locate tx-1,17:print "｡|:, "
1473:     beep  /*music "-a+c"
1474:     tw=0
1475:     wait(180)
1476:     locate tx-2,12:print "       "
1477:     locate tx-1,14:print "     "
1478:     locate tx-1,15:print "     "
1479:     locate tx-1,16:print "     "
1480:     locate tx-1,17:print "     "
1481:     if l<>1 then tn=0
1482:     ne=0
1483:     t_death=1
1484: endfunc
1485: /*
1486: /* 敵に技が効かなかった
1487: /*
1488: func no_damege()
1489:     locate x-2,12:print "           "
1490:     a=int(rnd()*10) mod 5 + 1
1491:     locate x,12:print hh(a)
1492:     wait(100)
1493:     locate x-1,12:print "             "
1494:     locate x-1,14:print "   ";c_d;c_l;c_l;c_l;"    "
1495:     h=0:at=0:z=0
1496: endfunc
1497: /*
1498: /* ケンタロウ倒れかける?
1499: /*
1500: func kenta_down()
1501:     locate x,15
1502:     print k1
1503:     locate tx+3,15:print."
1504:     locate x,14:print k(2)
1505:     locate x-2,15:print "  ｳ!!  ":beep /*music"-#c-#d"
1506:     for i=0 to 5000:next
1507:     locate x,12:print "   "
1508:     locate x,15:print k9
1509:     locate x,15:print k(1)
1510: endfunc
1511: /*
1512: /* スコア＆パラメータ表示
1513: /*
1514: func para_put()
1515:   locate 22,22:print "ﾜｻﾞ   ";wa(wk)
1516:   locate 22,20:print "ﾃｷ    ";na(1)
1517:   locate 1,22:print "SCORE   ";
1518:              print using"#####";sc*10
1519:   locate 1,20:print "ﾀｲﾘｮｸ   ";
1520:              print using"#####";kt
1521: endfunc
1522: func para_put2()
1523:   locate 28,22:print wa(wk)
1524:   locate 9,22:print using"#####";sc*10
1525:   locate 9,20:print using"#####";kt
1526: endfunc
1527: /*
1528: /* ウェイト
1529: /*
1530: func wait(wai)
1531: int i,j
1532:      for i=0 to wai
1533:        dm_key=inkey$(0)
1534:        v_disp()
1535:        if rt=1 then xt=xt+1
1536:      next
1537: endfunc
1538: /*
1539: /* データ初期設定
1540: /*
1541: func data_init()
1542:   c_u=chr$(&H1E)
1543:   c_d=chr$(&H1F)
1544:   c_l=chr$(&H1D)
1545:   c_r=chr$(&H1C)
1546:   bk(1)=" "+c_d+c_l+" "+c_u+c_l+"         "
1547:   bk(2)=c_d+" "+c_d+c_l+" "+c_u+c_l+"         "
1548:   bk(3)=c_d+c_d+" "+c_u+c_l+" "+c_d+c_l+"
1549: /*
1550:   bo(1)="/"+c_d+c_l+"/"+c_u+c_l+"--------"
1551:   bo(2)=c_d+"-"+c_d+c_l+"ｨ"+c_u+c_l+"--------"
1552:   bo(3)=c_d+c_d+"-"+c_u+c_l+"¥"+c_d+c_l+"--------"
1553: /*
1554:   k1="       "+c_d+c_l+c_l+c_l+c_l+"   "+c_u+c_u+c_l+c_l+"      "+
c_u+c_l+c_l+c_l+c_l+"   "
1555:   k2=" "+c_d+c_l+c_l+"   "+c_d+c_l+" "+c_d+c_l+c_l+"  "+c_d+c_l+
c_l+c_l+" "
1556:   k3="      "+c_d+c_l+c_l+c_l+c_l+c_l+c_l+c_l+c_l+c_l+"
"+c_d+c_l+" "
1557:   k4="  "+c_d+c_l+c_l+"  "+c_d+c_l+c_l+"  "+c_d+c_l+c_l+"
1558:   k6="      "+c_d+c_l+c_l+c_l+c_l+c_l+"       "+c_d+c_l+" "
1559:   k7="     "+c_u+c_l+c_l+c_l+"   "+c_d+c_d+c_l+c_l+c_l+"   "+c_d+
c_l+c_l+c_l+c_l+"   "
1560:   k9="     "+c_d+c_l+c_l+c_l+c_l+"    "
1561:   k10="       "+c_d+c_l+c_l+c_l+c_l+"   "+c_u+c_u+c_l+c_l+"      "
+c_u+c_l+c_l+c_l+c_l+"    "
1562: /*
1563:   k(1)=" [H "+c_u+c_l+c_l+c_l+" @ "+c_d+c_d+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+" /| "
1564:   k(2)="      "+c_d+c_l+c_l+c_l+c_l+c_l+"      "+c_d+c_l+c_l+c_l+c
_l+c_l+" @HH "+c_d+c_l+c_l+c_l+c_l+c_l+"  /| "
1565:   k(3)=" ---/ "+c_d+c_l+c_l+c_l+c_l+" / "+c_u+c_u+c_l+c_l+" //｣
"+c_u+c_l+c_l+c_l+c_l+" @ "
1566:   k(4)="@/"+c_d+c_l+c_l+"Hh"+c_d+c_l+c_l+"¥¥"+c_d+c_l+c_l+"_ ¥"
1567:   k(5)="@"+c_d+c_l+c_l+"｢¥¥"+c_d+c_l+"H"+c_d+c_l+c_l+"//"+c_d+c_
l+c_l+c_l+"/"
1568:   k(6)="L/HH¥"+c_d+c_l+c_l+c_l+c_l+c_l+"@L  ¥¥"+c_d+c_l+"¥"
1569:   k(7)="---H"+c_u+c_l+"Hh"+c_u+c_l+c_l+"@"+c_d+c_d+c_d+c_l+"|"
1570: /*
1571:   s(1)=" ロ "+c_d+c_l+c_l+c_l+" H｣ "+c_d+c_l+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+ " / ¥ "
1572:   s(2)=" 1ｪ> "+c_d+c_l+c_l+c_l+c_l+c_l+" H/ "+c_d+c_l+c_l+c_l+c_
l+" H¥ "+c_d+c_l+c_l+c_l+c_l+c_l+" / | "
1573:   s(3)=" 2 "+c_d+c_l+c_l+c_l+" H "+c_d+c_l+c_l+c_l+" H "+c_d+c_l
+c_l+c_l+c_l+" / ¥ "
1574:   s(4)=" 3/¥ "+c_d+c_l+c_l+c_l+c_l+c_l+" H "+c_d+c_l+c_l+c_l+" H
 "+c_d+c_l+c_l+c_l+c_l+" | ¥ "
1575:   s(5)=" 4 "+c_d+c_l+c_l+c_l+c_l+" /H¥｣ "+c_d+c_l+c_l+c_l+c_l+c_
l+c_l+" HHH "+c_d+c_l+c_l+c_l+c_l+c_l+" /H¥ "
1576:   s(6)=" o "+c_d+c_l+c_l+c_l+" H< "+c_d+c_l+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+" / ¥ "
1577:   s(7)=" O "+c_d+c_l+c_l+c_l+" H[ "+c_d+c_l+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+" /|  "
1578:   s(8)=" D "+c_d+c_l+c_l+c_l+" H｣ "+c_d+c_l+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+" | ) "
1579:   s(9)=" 0 "+c_d+c_l+c_l+c_l+" H｣ "+c_d+c_l+c_l+c_l+c_l+" H "+c_
d+c_l+c_l+c_l+c_l+" | ¥ "
1580:  s(10)=" o "+c_d+c_l+c_l+c_l+c_l+" <H｣ "+c_d+c_l+c_l+c_l+c_l+" H
 "+c_d+c_l+c_l+c_l+c_l+" < ¥ "
1581:   s1="     "+c_d+c_l+c_l+c_l+c_l+c_l+"     "+c_d+c_l+c_l+c_l+c_l
+c_l+"    "+c_d+c_l+c_l+c_l+c_l+c_l+"    "
1582: endfunc
1583: /*
1584: /* 変数初期化＆背景描画
1585: /*
1586: func init_back()
1587: int hx,hy
1588:   cls
1589:   jk=1:ro=1:n=1:x=33:tx=6:ts=300:rt=0:cg=0
1590:   t_death=0:death=0
1591:   kentarou=3
1592:   ya=0
1593:   for i=0 to 39
1594:      locate i,18:print "+";
1595:   next
1596:   for i=0 to 30
1597:     hx=int(rnd()*100) mod 40
1598:     hy=int(rnd()*100) mod 11
1599:     locate hx,hy:print "."
1600:   next
1601:   wk=1:locate 12,3
1602:   print "*  ";c_u;c_u;"*  ";c_d;c_d;"*";c_d;c_d;c_d;c_l;c_l;"*
";c_d;"*";c_d;c_d;"   .";c_l;c_u;"*    *"
1603:   l=l+1
1604: endfunc
1605: /*
1606: /* 北斗の男プロローグ
1607: /*
1608: func opening()
1609: int i,j
1610: int aw
1611: str k
1612: str opn[255]="2000ﾈﾝ ﾉ ﾚｷｼ ｦ ｷｻﾞﾐ++ｳｹﾂｶﾞﾚﾃｷﾀ ｵｿﾙﾍﾞｷ++ｱﾝｻﾂｹﾝ ｶﾞ ｱ
ｯﾀ.++ｿﾉ ﾅ ｦ ｢ﾎｸﾄ ｼﾝｹﾝ｣!!++ﾃﾝｸｳｳ ﾆ ﾂﾗﾅﾙ 7ﾂﾉ ﾎｼﾉ++ﾓﾄ ｲｯｼｿｳﾃﾞﾝ ﾉ ﾎｸﾄｼﾝｹﾝ++ｦ
ﾒｸﾞｯﾃ ﾋｹﾞｷ ﾊ ｸﾘｶｴｻﾚﾙ.......+++｢ｾｲｷﾏﾂ ｷｭｳｾｲｼｭ ﾃﾞﾝｾﾂ｣ ﾖﾘ ﾊﾞｯｽｲ ....."
1613:   cls
1614:   j=3
1615:   locate 10,3
1616:   i=0
1617:   k=""
1618:   while i<197 and k=""
1619:     if mid$(opn,i,1)<>"+" then {
1620:        print mid$(opn,i,1);
1621:        for aw=0 to 4:v_disp():next
1622:     } else {
1623:        j=j+1:locate 10,j
1624:     }
1625:     i=i+1
1626:     k=inkey$(0)
1627:   endwhile
1628:   for i=0 to 5000:next
1629: endfunc
1630: /*
1631: /* 操作説明
1632: /*
1633: func sousa()
1634:   cls
1635:    print "         <<<[ ﾎｸﾄ ﾉ ｵﾄｺ PART1 ]>>>"
1636:    print " ｹﾝﾀﾛｳ ｦ ｱﾔﾂｯﾃ 10ｼｭﾙ ﾉ ﾃｷ ﾄ ﾀﾀｶｯﾃｸﾀﾞｻｲ."
1637:    print "      ｷﾐﾊ ﾎｸﾄｼﾝｹﾝ ﾃﾞﾝｼｮｳｼｬ ﾆ ﾅﾚﾙｶ !!"
1638:    print
1639:    print "                KEY ﾉ ｾﾂﾒｲ"
1640:    print
1641:    print "  ｢~｣                ｢~｣
1642:    print "  |4| .... ﾏｴ ｲﾄﾞｳ  |6| .... ｳｼﾛ ｲﾄﾞｳ
1643:    print "  ｢_｣                ｢_｣
1644:    print "  ｢~｣                ｢~｣
1645:    print "  |8| .... ﾏｴ JUMP  |2| .... ｳｼﾛ JUMP
1646:    print "  ｢_｣                ｢_｣
1647:    print "  ｢~｣ ｢~｣
1648:    print "  |X|&|V| ....ﾜｻﾞ ｦ ｶｴﾙ"
1649:    print "  ｢_｣ ｢_｣ 1)ｾｲｹﾝﾂﾞｷ    2)ｱｼｹﾞﾘ"
1650:    print "           3)ﾋｺｳ ｦ ﾂｸ (3ｼｭﾙｲ)"
1651:    print "           4)ﾎｸﾄ ｻﾞﾝｶｲｹﾝ 5)ﾎｸﾄ ﾋｬｸﾚﾂｹﾝ"
1652:    print "           6)ｺｳｼｭ ﾊｶﾞﾝｹﾝ 6)ﾎｸﾄ ｶｲｺﾂｹﾝ
1653:    print "           8)ﾎｸﾄ ｼﾞｭｳﾊｻﾞﾝ"
1654:    print "｢~~~｣                ｢~｣"
1655:    print "|SPC| .... ﾜｻﾞ ｦ ﾀﾞｽ |C| .... ﾆｼ ｼﾝｸｳﾊ"
1656:    print "｢___｣                ｢_｣ (ｽﾍﾟｰﾄﾞ ﾆ ﾂｶｳ)"
1657:    print
1658:    print "             (C) LUM SOFT"
1659:    print "   ARRANGED BY YAMATO SOFTO & NISHIYAMA";
1660: /*
1661:    locate 12,22:print "HIT ANY KEY !!"
1662:    i=0
1663:    while i<2000 and inkey$(0)="":i=i+1:v_disp():endwhile
1664:    if i<2000 then i=0
1665: return(i)
1666: endfunc
```

▶バイク(NC30)で事故りました。みなさンも「毒電波」には気をつけましょう。
藤田　康一(23)静岡県

新製品紹介

倍クロックアクセラレータH.A.R.P

Taki Yasushi 瀧 康史

ついに発表されたH.A.R.Pですが，発表後もなかなか平坦な道ではないようです。動作機構を推測しつつ，どのような部分が高速化されるのかを見てみましょう。

●

ついに出たH.A.R.P

待ちに待ったH.A.R.P。とにかく，待った人は待ったよね？　最近はSX-WINDOWなどをはじめとする重めのソフトがたくさん出てきているし，近ごろのゲームはみんな10Mショックなものばかり。「ああ，もう10MHzじゃ耐えられなーい」という人もたくさんいるだろう。編集部にも，スーパーストリートファイターIIは10MHzマシンでもH.A.R.Pを使用すれば速くなりますか？　などという質問がたくさんきている。期待もひとしお。

というわけで，いつもより，ちょっと厳しめにH.A.R.Pをレビューしてみよう。

取り付けに関して

取り付けはCPUをはずして代わりにH.A.R.Pの基板を装着することで行われる。基板の陰になってピンのある位置がわかりづらいので，こういうことをやったことがない人にはちょっと難しいかも。

X68000初期型はスロットの真下にCPUがある。写真の試作バージョンではICソケットがついているためスロットがひとつ潰れてしまったが発売されるバージョンは問題ないということ。スロットは2つとも使えるぞ。ボードは非常に薄く作られているのでEXPERTなどでも装着したままシールド板を取り付けることが可能になっている。

ただ，初期型の本当に初期の型はCPUのすぐ隣にあるROMがEP-ROMなので，H.A.R.Pの基板が少しぶつかってきっちり差し込むことができない。EP-ROMは普通のマスクROMよりも厚みがあるからだ。

EP-ROMの初期型はいわゆる超初期型というやつで，ROMのバージョンが古い。このEP-ROMに差し替える新しいバージョンのマスクROMがシャープから補修部品で購入できる（私はツクモ電機でその昔購入した。いまも売ってるかわからないけど）。マスクROMに替えると，動かなかったソフトが動くことが「ときどき」あるので，この機会に取り替えてみるのもよいかもしれない。

動作原理

簡単にいえばH.A.R.Pは2倍速アクセラレータだ。入力されたクロックを倍速化してCPUを駆動する。外部バスは従来のままで，内部を2倍にしてアクセスしている。これは，インテルCPUのDXとDX2の関係に似ているな。

だから2倍といっても，実速度が2倍というわけではなくて内部クロックが2倍というわけ。外部がそのままの分，その調停があるから，実際のパフォーマンスは低下してしまう。では結果的にどのくらい速くなるのか？　実際に試してみた。

まずは，H.A.R.PをつけてみてSX-WINDOWなどを試してみる……多分速くなっているんだろうけど……全然わからないぞ？　エディタなどを適当に使ってみたけれど，効果はさっぱりわからない。どうやら試すものが悪いのか，H.A.R.P.を入れたことによって，体感的にわかるものがなかなかみつからない。

Oh！Xでよく使われているStanfordベンチマーク（XC ver.2.0）で試したところ，整数で10MHz時の1.06倍，実数で1.08倍だった。一方，Dhrystone（GCC）では1.13倍。結果的にだいたい1割ぐらいのスピードアップをしているようだ。

ゲームなども試してみることにした。アクションゲームは10MHzでも動くように作られているうえ，垂直同期を見ているので差は出ない。あえていえば，ところどころで重くなったのが，少しは解消されたかな？　という程度。

PCM8などを使ったときにパフォーマンス的に1割上がった分，あまり重くならないかなといったところか。

そこでXVI登場以前のゲームで速度調整のしてないスタークルーザーを試してみた。内部計算が多そうな分，速さが出てくるかと思ったものの，結果は大したことなし。オープニングが始まってから終わるまで，10MHzマシンでは3分15秒ぐらいだったのが，H.A.R.P装着では3分10秒程度。スタークルーザーのオープニングも，ところどころで同期を取っているところがあるだろうが，それをおいても，ちょっとな。だってXVIだと2分20秒程度なんだもん。

そこでまったく同期を取ってないSION2のオープニングにかかる時間を計ってみた。起動してからオープニング終了まで，10MHzマシンでは1分22秒だったものが，H.A.R.P実装機種では1分13秒で終了した。やっぱり1.13倍程度か。ちなみに，XVIは51秒，X68030（i-cache：on d-cache：on）は20秒だった。ただし1.13倍といってもH.A.R.P装着機と10MHzマシンを2台並べてみると，かなりスピードアップをしていることがわかる。

互換性に関していえば，同じMC68000を使っている分，石上さんの030アクセラレータなどと違って完璧なはず。完璧なはずなんだけどなあ……。

なぜあまり速くならないか？

内部クロックが20MHzなのになぜあまり速くならないんだろう？　そう思う読者もいるだろうから，私が予想を含めて考えてみた。正確な資料をもらっているわけではないので間違えているかもしれないが，だいたいはこれで当たっているだろう。

CPUはまず20MHzで動く。しかしバス周り（本体側）は10MHzだから，これを調停することが必要になる。アクセラレータとして働く手前，見かけ上はあたかも10MHzのMC68000として動かなくてはいけないわけだ。したがって，CPU内部で演算するときには20MHzでフルに動いても構わないが，外部バスにアクセスするときには，10MHz相当に落とさねばならない。

さて，68000CPUはどんな命令も必ず4クロックはかかる。これはNOP命令が4クロックであることから明らかだろう。

命令をメモリから読み出し，それがどういう命令であるか解析し，アドレス解析などを行って実行に移るという仕組みだ。このうち，命令の解析などはほとんど時間がかからないため，結局はこの4クロックは，ほとんど外部のメモリとのやりとりで消費

される。MC68000というCPUはオンチップキャッシュがないため，外部バスの終了時間を地道に待つからだ。

ということは，H.A.R.Pが内部的に20MHzで動いたにしても，結局は外部バスとの調停で待たなければいけないので，この基本4クロックは変わらない。

4クロックの命令にはどのような命令があるだろうか？　まずNOPに始まって，実はMOVE命令のレジスタ間転送は，4クロック命令だ。したがってレジスタ間MOVE命令は高速化されない。おそらく，別売されるER10Sとあわせれば，この基本アクセスが3クロックになって高速化されるのだろう。そうなればどの命令も速くなり，結果的に平均パフォーマンスは1.4倍ぐらいにはなるんだろう。

ともかくH.A.R.P単体では基本4クロックの2バイト命令は高速化されない。

次に実験としてMOVE命令にイミディエイト指定があるものでmoveqにならないものを試してみた。結果はH.A.R.Pを装着すると1クロックほど遅くなっていることが判明。この手の命令は，命令コード1ワードの先に2ワードのデータが存在する。この2ワードのデータを引き込むときに，なぜかウェイトが入ってしまうらしい。なんらかの外部バス間の調停をするときにミスするのであろう。しかしながらうまく作れば遅くなるなんてことはありえない。これはおかしいな。

BSRなどのジャンプ命令も，この手の命令だ。先の2ワードデータがアドレスであるにしろ，この取り込みが遅くなるのは，かなりのネックになっているに違いない。

では，速くなる命令はなにかというと，たとえば，EXG命令。この命令は命令コード1ワードだけの命令で，6クロックを消費する。最初の4クロックは外部バスアクセスなのでH.A.R.Pでも結果は同じだが，このあとの2クロックは完全に外部バスは開いていて，CPU内部で2クロックを消費するものだ。結果，この2クロックは内部の20MHzで実行されるため，10MHz時相当の1クロックに相当する。ということはEXG命令は6クロック命令から，H.A.R.P実装により5クロック命令になるということになるわけだ。

H.A.R.P実装によって速くなる命令は，このほかにもMOVE to CR命令など。この命令は1ワード命令なのだが，ソースがデータレジスタのときでも，12クロックも消費する。基本クロックが4クロックであることを考えると，12−4＝8の分は内部で20MHzで実行され4クロックで処理されることになり，結果的にこの命令は4＋4＝8クロックで処理される。このような命令はシステムレジスターデータレジスタ間命令のほか，movem命令やlink/unlk命令などがある。

速くなる命令で，いうまでもないのが演算命令。add,sub,cmp,mul,div,asrなどのすべてが速くなる。ただしレジスタ間の場合，イミディエイトとの引き算や比較をすると少し遅くなってしまうかもしれない。

また（a0）などの相対アドレスを使った命令を実測してみたところ，これも4クロックと変わらないようだった。実測に使った命令はmove.l(a0)，d0で，1ワード命令ながら12クロックも消費する命令。やはり12−4＝8クロック分は外部バスとのやりとりに，ほぼすべてを使われているのだろう。

結果，H.A.R.Pを利用して速くなる命令は1ワード命令でアドレッシングモードにアドレスレジスタ直接指定（(a0)などの表記）を利用しないもの，ということになる。もちろん，move.l d0,d1のような4クロック命令は除く。実験はしていないが，おそらく(a0,d0.l)　などという命令は，内部でa0＋d0.lの計算を行っているから多少速くなるだろう。要はレジスタ間で行うようにオプティマイズされていればよいので，GCCでコンパイルしたようなソフトは速くなるかもしれない。

H.A.R.P　29,800円（税別）

まとめ

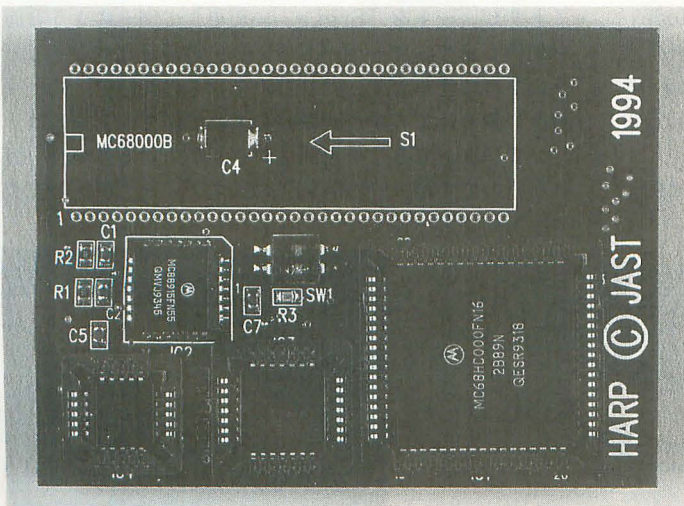
編集部にあるバージョンは製品版ではないため，製品版とは異なるかもしれない。ただ，オペランドのほかにデータを含んだ命令が遅くなるという仕様は発売間近ということもあり製品版でも直らない可能性が高いだろう。

H.A.R.Pで速くなるかどうかは，結局のところプログラム中にいかに速くなる命令が散りばめられているか，遅くなる命令が少ないかによるわけだ。問題は，平均して確実に1割速くなると見るか，1割しか速くならないと見るかだが……。これに29,800円をかけるか否かは各人の判断に委ねられる。

まだ見ぬER10S（72ピンSIMM使用のメモリボード）がカタログに記載されているとおりのスペックならば，すべての命令は1クロック必ず減り，おそらくオペランドのほかにデータを含んだ命令の仕様が解消されるだろう。そうなれば，メーカーがいうようにXVI程度にはスピードアップするかもしれない。

ターゲットはやっぱり，いまだにメモリを増設してない人か？　X-SIMMが出る前ならばいけたかもしれないが，あれが発売された以上，もはや，2Mバイト，10MHzユーザーはかなり減ってきているだろう。コンパクトXVIも8万円強で買える昨今。どうなることやら……。クロックアップ改造に手を出したけど12MHzくらいが限界という人には朗報なのかな？

㈱ジャスト　☎03(3706)9766

DōGA CGアニメーション講座 ver.2.50……………………(第20回)

マニュアルを手に再チャレンジ！(後編)

プロジェクトチームDōGA
かまた ゆたか

前回の続きの実践的なモデリングについて解説したのち，初心者にとってのもうひとつの挫折の原因について考えてみましょう。また，アマチュアCGA学会からの最新技術のレポートもあります。

はじめに

さて前回は，マニュアルを手にした初心者の方にとって，まず何から始めるべきかということで，マニュアルの読み方，そして初心者が挫折する最大の原因といわれるモデリングについてお話ししました。しかし，ページの都合で，モデリングの解説の途中で終わってしまいましたので，まずはそこから再開しましょう。

今回もCAD以外のツールも積極的に使って，より実践的にモデリングします。

パーツBの制作(回転体)

先月の話ですから，なにをモデリングしていたのか忘れてしまいましたね。GENIEで使うパーツとして，図1のようなブースターを制作していたんですよ。それを図2のように分解して，いちばん複雑なパーツAは先月完成させました(写真1)。ついでに，パーツEの部分も終わっています。

ということで，パーツBの制作から入りましょう。とはいっても，ただの噴射口，CADの回転体で1発です。回転体とは，ろくろで回転しながら作った壺のような形状(輪切りにしたとき断面が円になる形状)で，回転軸と，輪郭を指定してやれば，その輪郭を軸回りにぐるっと1回転して生成します。

準備としては，CADを起動し，「Attribute Mode」/「Attribute登録」で，GENIEで使えるアトリビュート名を登録しておきます。今回は「BodyL」と「EngLg」の2つでよいでしょう。そして<Panel 1>の「atr no」のところを「BodyL」にするのを忘れないように。

パーツAとパーツBをくっつけるときは，どうせ拡大縮小するのですから，パーツBをどんな大きさで作ってもかまいません。しかし，そんなに大きく作っても仕方がありませんから，「scale」を2か1にしておきましょう。

「Edit Mode」に入ったら，制作開始です。「回転体作成」に入ると，デフォルトで角数が「12」になっています。つまり，断面が完全な円ではなく，正12角形で近似するということです。別にそのままでもいいのですが，「可能な限り面数は減らせ」というのがモデリングの基本ですから，この「12」を右クリックして，「8」か「6」にしてください。

回転体を作るときは，まず回転軸を指定します。この噴射口の回転軸はX軸でよいので，平面図か側面図でX軸上の任意の2点を指定します(図3)。

次に，噴射口の輪郭を描きます。面数を少なくするためにも，3点指定すれば十分です(図4)。輪郭が確定すれば，リターンで回転体を発生します(図5)。

結構簡単にできますね。昔はこの回転体は，面数がすぐ多くなるという理由で嫌われていたのですが，最近はメモリも増え，処理速度も向上していますので，どんどん使ってかまわないでしょう(でも角数は控えめにね)。

ただ，この回転体作成には，計算誤差が妙に大きいという欠点があります。たとえば，今回作った噴射口は，左右も上下も対称形になっているはずですが，たぶんどこかの頂点が微妙にずれているでしょう。

ところで，いま作られた噴射口は，筒状のままなので，アトリビュートを「EngLg」に変更して図6のような面を制作します。キーボードの「8」の最近点でなぞるようにすれば簡単です。ただし，先ほどの問題で，同一平面上に載るはずの頂点が微妙にずれていて，1つのポリゴンでフタができない可能性があります。その場合，情けないのですが，一度SAVEして，エディタで頂点の座標

図1 ブースターのデザイン

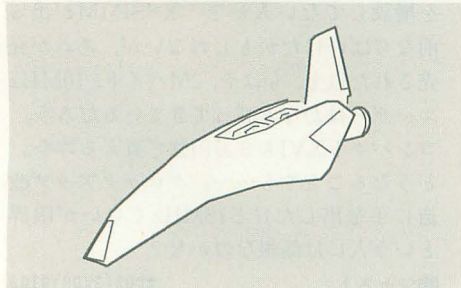



図2 パーツに分解

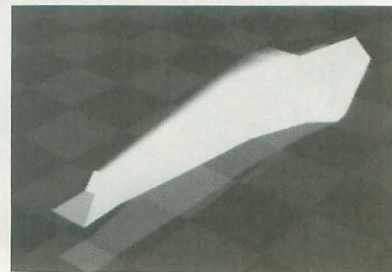
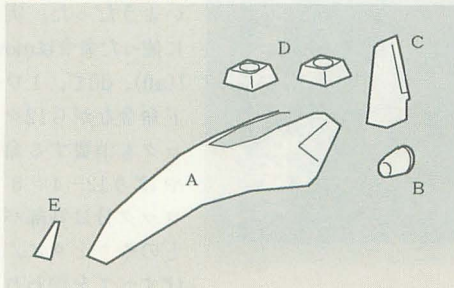



写真1 先月号で制作したパーツA

図3　回転軸の2点の指定

図4　輪郭の確定

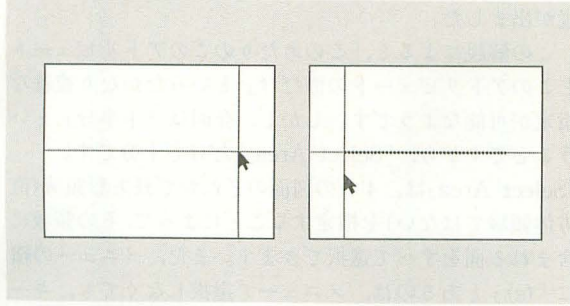
図5　回転体の発生

を修正してください(形状ファイルをエディタで修正するなんて，一見非常に大変そうですが，やってみると意外と簡単です)。

図6　フタをする面を制作

以上でパーツBのCADでの作業は終わりですので，「b1.SUF」としてSAVEして終了します(写真2)。

最後にこの「b1.SUF」に対して，SHADEで曲面化してやります。

SHADE b1.SUF /Ob2

出力された「b2.SUF」がパーツBです。

パーツCの制作

解説することは特にありません。CADの側面図で，1，2面から成る図7のような垂直尾翼を作って，「c1.SUF」としてセーブしてください(写真3)。

パーツDの制作(MODEL.Xの試用)

さて，最後のパーツですが，この部分はすでにあるデータを修正して使用するという方法をとってみましょう。すでにあるデータとは，もちろんGENIEのデータベースです。まず，GENIEの「parts¥タンク¥TK05.SUF」を現在使用しているワークディレクトリにコピーしてください。

この「TK05.SUF」にどのような修正を加えるかというと，パーツDは下半分が隠れているので，このへんの見えない面を全部削除してしまうだけです。「可能な限り面数は減らせ」です。

面の削除は，もちろんCADでもできますが，その場合1面ずつ削除しなければいけません。その点，新しいモデラであるMODEL.Xは，複数の面をまとめて削除する機能がありますので，今回はこ

図7　パーツC

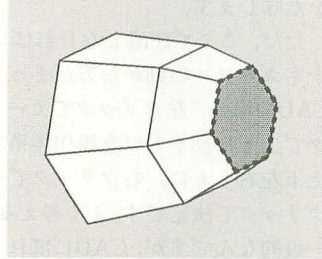
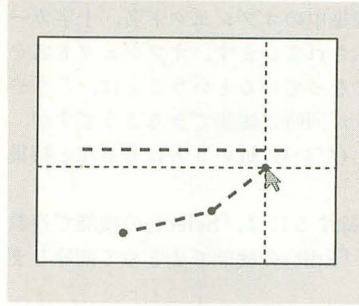
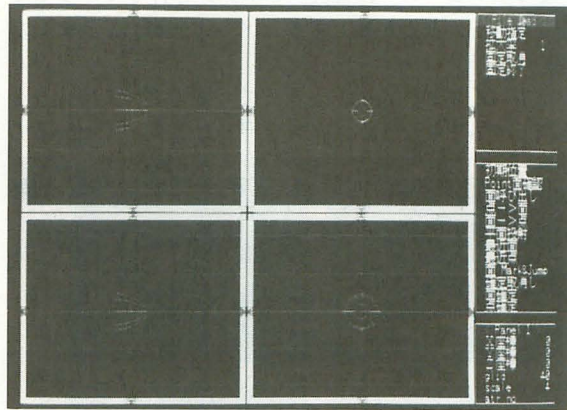
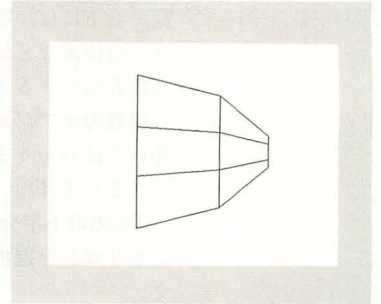



写真2　パーツBを作る

のMODEL.Xを使ってみましょう。

とはいっても，実は私はまだMODEL.Xを使ったことはないんです。追加マニュアルにもMODEL.Xの解説はありませんので，使い方がわからないのは，皆さんと大差ありません。今回は，必要最小限の使い方だけにして，詳しくは，もう少し研究してから，また近いうちに取り上げたいと思います。

さて，TK05.SUFを読み込みながらMODEL.Xを起動するには，

MODEL TK05⏎

とします。ただし，昔のSLASHのモデラの名前もMODEL.Xだったので，それをインストールされている方は，パスの指定によってはそちらが起動してしまう可能性があります。ご注意ください。

写真4が，起動時の画面です。作者が小林君なので，Ko-WINDOWの雰囲気が漂っていますね。4つの図面はCADと同じですが，各機能は，上部のプルダウンメニ

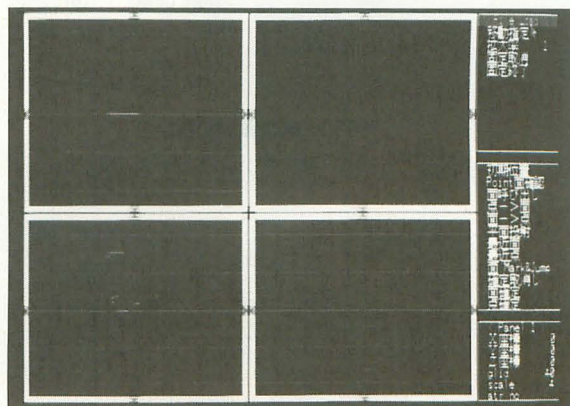
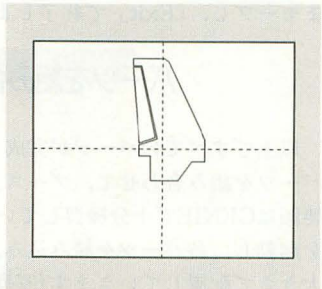



写真3　パーツCを作る

ューを開いて使用します。右上にはアトリビュートの一覧。右下には，現在編集中のオブジェクト名，十字カーソルの位置などが表示されています。オブジェクト名を複数表示できるようになっているということは，どうやら複数のオブジェクトが同時に編集できるようですが，使い方はわかりません(だから，近いうちにちゃんと特集をするってば)。

TK05の下半分を削除するには，「Select」の機能で複数のポリゴンを選択し，「Edit」の機能でまとめて削除します。

まずメニューの「Select」でボタンを押してみましょう。すると，なんか難しそうなメニューが出てきました(図8)。

いちばん下が「Help」となっているので，まずこれを選択してみましょう。ボタンを押したまま，「Help」まで

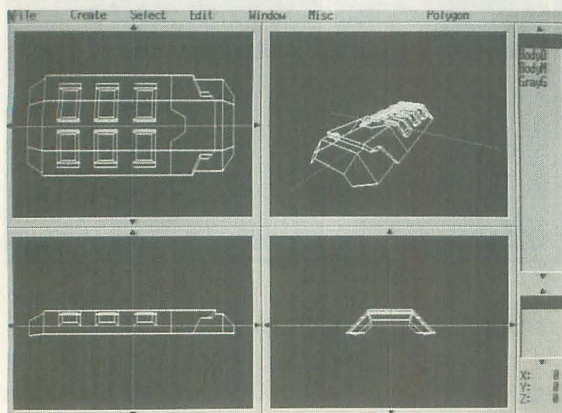



写真4　MODEL.X起動時の画面

図8　Selectメニュー

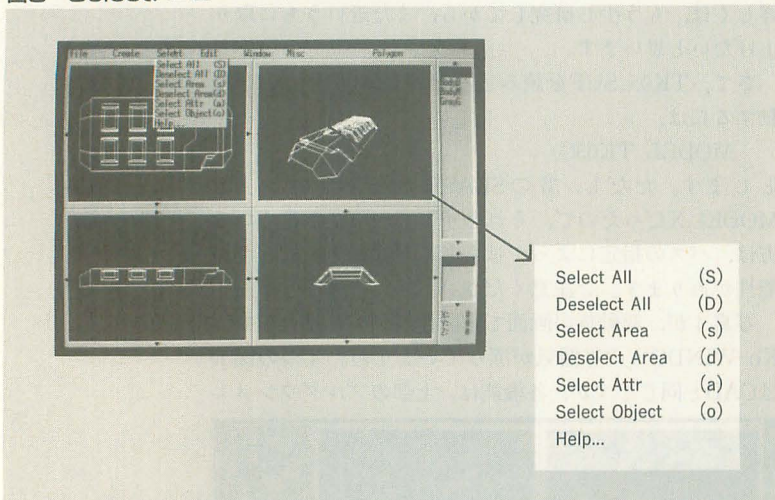



図9　選択領域の指定1

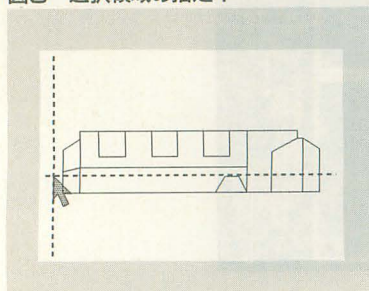
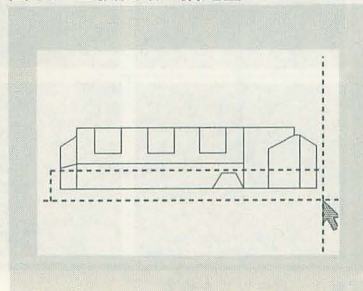



図10　選択領域の指定2

持っていき，ボタンを離します。はい，ちょっとした解説が出ました。

この解説によると，「このあたりのこのアトリビュートとこのアトリビュートの面だけ」といったかなり複雑な指定が可能なようです。しかし，今回は「下半分」ということですから，「Select Area」だけで十分です。「Select Area」は，4つの図面のどれかで長方形領域(直方体領域ではない)を指定することによって，その領域に含まれる面をすべて選択できます。また，メニューの横に「(s)」とあるのは，メニューで選択しなくても，キーボードの「s」を押すことで，このモードに入れることを意味します。

ただ，ここで注意しなければいけないのは，CADとは十字カーソルの動かし方がまったく異なるという点です。CADの場合，左クリックでカーソルが左右に，右クリックで上下に動くという操作体系でしたが，MODEL.Xは，上下左右ともに，右クリックで十字カーソルが動き，左クリックで決定です。よく考えると，MODEL.Xのほうが一般的なんですが，CADに慣れている方はかなり混乱すると思います。

具体的な操作は，まず側面図(正面図でも可)で，図9の位置にマウスカーソルを持っていき，右クリックで十字カーソルを移動させ，左クリックで決定します。次に図10の位置にマウスカーソルを移動し，右クリックで十字カーソルを持ってきて(このとき，選択される領域が黄色い長方形で表示される)，左クリックで領域の決定です。虹色に輝いている(？)のが選択された面です。

この操作方法には，ちょっと慣れが必要ですね。思っていたところがうまく選択できなかった方は，「Deselect All(D)」で何も選択していない状態からやり直すか，「Deselect Area(d)」で間違って選択してしまった面を除外して，SHIFTキーを押しながら再び「Select Area(s)」を実行して，足りない面を加えてください。

うまく下半分の面が選択できたならば，選択した面が虹色に輝いている状態で，「Edit」の「Delete(DEL)」を実行してください。選択されていた面がすべてなくなってしまいました。なかなか便利でしょう？

作業は以上で終わりですので，SAVEして終了してもいいのですが，その前にキーボードの「－」を押してみましょう。何が起こるかは，実際にやってみてのお楽しみです。ということで，このMODEL.X，使い方がわかれば結構使いものになるかもしれませんね。

おっと，いま気がつきましたが，このMODEL.Xには，名前を変えてSAVEという機能がないようです。仕方がないので，「File」の「Save」でとりあえず元の名前のままセーブし，「Exit」で終了しましょう。

パーツを組み合わせる

以上ですべてのパーツが完成しましたので，それらのパーツを組み合わせて，ブースターを完成させましょう。要領はGENIEで十分練習しているはずです。まず，FFEを起動し，各パーツを読み込み，適当な位置に，適当な大きさで配置していきます(写真5)。

DōGA

パーツBは，あまりパーツAに近づけるとくい込んでしまうのでご注意ください。パーツCは必要に応じてY軸を回転させるとよいかもしれません。パーツDもパーツAにくい込まないように，ちょっと小さめに，奥のほうに設定するのがよいでしょう。パーツCとパーツDがくい込むのはデザイン的にいって問題ありません。

完成しましたら，ファイルをセーブします。「NEW1」とでも名前をつけておいてください。そして終了します。GENIEの場合，FFEを終了すると，自動的に形状を生成していましたが，今回は「NEW1.FSC」に基づいて形状を生成するように指示を出してやる必要があります。これには，KAMA.X を使用します。

KAMA NEW1⏎

これで，「NEW1.FSC」から「NEW1.SUF」が生成しました。以上で，ブースターの完成です(写真6)。しかし，FFEのワイヤーフレームの画面と，レンダリングした画像とはかなりイメージが異なりますので，AUTO.Xで作画させてみましょう。

AUTO /A2 /G NEW1.SUF GENIE.ATR⏎

でアニメーションが作られます。ただ，AUTO.Xが自動的に生成するフレームソースが「NEW1.FSC」になるため，

「NEW1.FSC」が既に存在します。どうしますか？

というWARNINGが出ます。この場合，「3：フレームソースファイル名を変更する」を選択(3を入力)し，試しにつくるフレームソース名として「TEST」などを入力してください。

アニメーションを見て，バランスなどに問題があったら，FFEを起動し，「NEW1.FSC」を読み込んで修正してください。

いかがでしょうか。最初のデザイン画とは多少異なりますが，最初にもいったように，あまり原画にはこだわらず，いろいろ楽しみながら作業することが大切です。また，修正することを考えて，途中の形状をいくつか残しておいてください。ちょっとした修正のために一からやり直しなんて効率が悪いですから。

全体の変形(BOXTRANS)

先ほどの写真6は，だいたい思いどおりの形になりましたが，いつもそううまくいくわけではありません。完成した形状がどうもバランスが悪い，後部はもうちょっと太くとか，前部はもうちょっと長いほうがいいなどということもあるでしょう。FFEで修正できればよいのですが，パーツA自身を修正するとなると，OUTLINEの元の3面から作り直さないといけません。ちょっと面倒ですね。

このように，最後の最後で全体の形を変形するには，BOXTRANSが有効です。ただし，最初にいっておきますが，このツールは，面数が増える，使いにくいという大きな欠点があります。まぁ，がんばって使ってみましょう。

BOXTRANSとは，物体の一部(もしくは全部)を直方体で包み，その直方体の各頂点を移動させることで，直方体で包まれた部分の形状を変形させるというツールです。ただし，直方体の座標や，各頂点をどの方向にどれだけ移動してやるかを視覚的に指定できるような機能はまったくなく，エディタでそれらの座標値を記述したコマンドファイルを作らないといけません。詳しい機能については，追加マニュアルの第3章「新ツール一覧」(A-1)をご覧ください。

今回は，ブースターの後部を，図11のように変形してみましょう。後ろの裾を後方に広げ，後ろの上部を上に持ち上げるような感じです。

そのためには，まず，図12のように，後部を包み込むような直方体の座標を求めなければいけません。それには，CADを起動してNEW1.SUFを読み込み，実際に直方体を作ってみるのがよいでしょう(写真7)。そして，その頂点の座標をメモしてください。8頂点すべてをメモする必要はありません。左前上と右後下のように，対角線上の2頂点で十分です。私の場合の直方体の座標は，(-80, 340, 320)と(-1160, -340, -40)でした。

次に，そのCADの画面上で，どの頂点をどれだけ動かすかという値も求めてください。目分量でおおよその値を求めればよいでしょう。そして，その値もメモしてください。今回私は，(-1160, 340, 320)を(-1160, 120, 440)に，(-1160, 340, -40)を(-1310, 560, -120)に変形させます(右半分も同様：図13)。

これらの座標値がわかれば，コマンドファイルを書くのは簡単です。変形を指定するコマンドには，vectorとboxtransの2種類があります。vectorは，直方体の各頂点がどれだけ移動するかの移動量を記述するのに対して，boxtransは，各頂点がどこに移動するかという具体的な座標を記述します。つまり，どちらを使っても同じ結果が得られるのですが，今回の場合，前の4頂点は動かない，つまり移動量が0ということがはっきりわかっているので，vectorのほうを使ってみました(リスト1)。

vectorでも，boxtransでも，注意しなければいけないのは，頂点を記述する順番です。追加マニュアルの図のように，1行目は左前上の頂点，2行目は左後上，……，

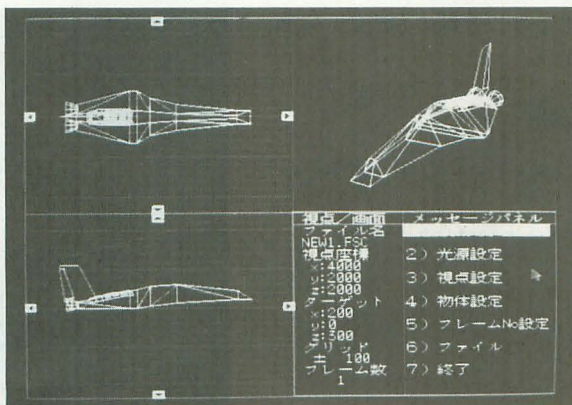



写真5　各パーツを配置する

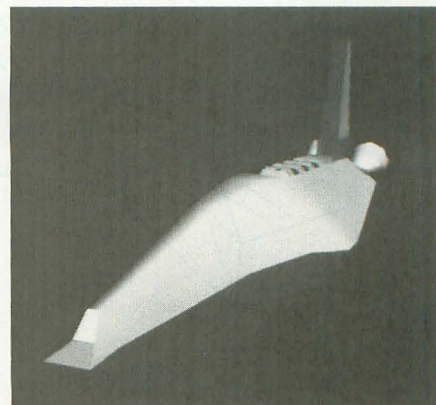



写真6　完成したブースター

図11 BOXTRANS の変形

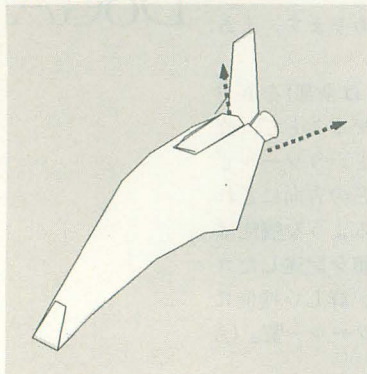



図12 直方体で包む

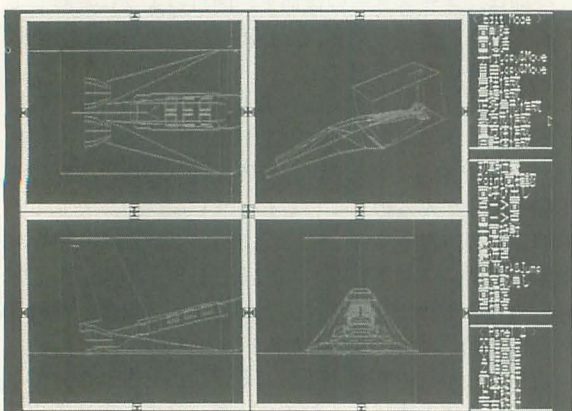
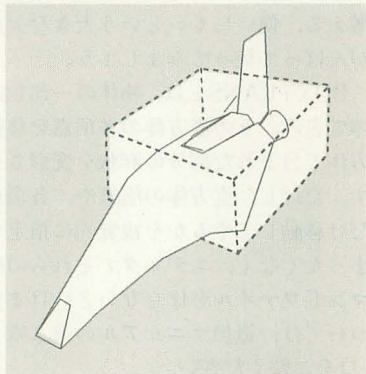



写真７ 変形するための座標値を求める

という順番を守らないと，とんでもない変形をします。

コマンドファイルの記述ができたら，

BOXTRANS NEW1.SUF /fhen1.txt /ohen1

のように実行します。/fでコマンドファイルを，/oで出力ファイル名を指定します。処理には意外と時間がかかるという点と，出力ファイルのオブジェクト名は入力ファイルのオブジェクト名になるという点にご注意ください。

変形した結果は，写真８です。左側がオリジナルで，右側がBOXTRANSの処理をしたものです。ちょっとわかりにくいのですが，よく見るとノズルがひしゃげているのがわかると思います。

GENIEへの登録

完成したブースター「NEW1.SUF」をGENIEに登録して，X01号に加えてみましょう。登録の方法は，以前GENIEの解説で詳しくやっていますので，ポイントだけをまとめます。

まず，GENIEの「parts」というディレクトリの中に，たとえば「追加」というディレクトリを作り，その中に「NEW1.SUF」をコピーします。次に，「sparts」というディレクトリにも「追加」というディレクトリを作り，その中にOUTLINEを実行させるために制作した３面だけの形状をコピーします。これがシンプル版の形状というわけですが，ファイル名を「NEW1.SUF」に変えることと，オブジェクト名も「NEW1」にしておくことを忘れないでください。GENIEを起動すれば，ちゃんと「追加¥NEW1」というパーツが加わっているはずです。

写真９がX01号にブースターを加えたものです。いかがでしょうか？ 少しはカッコよくなりましたか？

第2の挫折

以上のように，「CADなんかでモデリングができるわけない」と挫折していた方も，

・制作に入る前に，方針を立てる
・複雑な物体を簡単なパーツに分けて制作する
・CAD以外のツールを活用する

という３つのコツを押さえることで，なんとかなりそうだと感じてもらえたと思います。しかし，しばらくするともうひとつの挫折が待っています……。

CGA大学の「CAD基礎実習」が終わると，専門課程に進みます。最初の「ATR基礎実習」は，操作自体は簡単です。各パラメータの意味を理解し，適当な質感を表現するのは難しいですが，そこまで追求する方は，教養課程の「色と材質の概念」(T-59)と，修士課程の「アトリビュート研究」(T-223)もあわせて勉強してください。

専門課程の大部分を占める「FFE基礎実習」は，非常に面倒くさいのですが，GENIEで遊んだことのある方にとって，すでに手慣れた作業です。マニュアルで使用しているのは，かなり古いバージョンのFFEですので，操作方法に若干差がありますが，特に問題ないでしょう。

そのほかの「SIMPL基礎実習」「WIREVIEW基礎実習」「REND基礎実習」などは，もういちいちPESから実行せず，コマンドラインから入力してください。「TELOP基礎実習」は選択科目ですから，とばしても差し支えありません。

図13 直方体の変形

リスト１

```
box ( -80 340 -40 -1160 -340 320 )
vector (
        0    0    0
        0 -220  120
        0  220  120
        0    0    0

        0    0    0
     -150  220  -80
     -150 -220  -80
        0    0    0
```

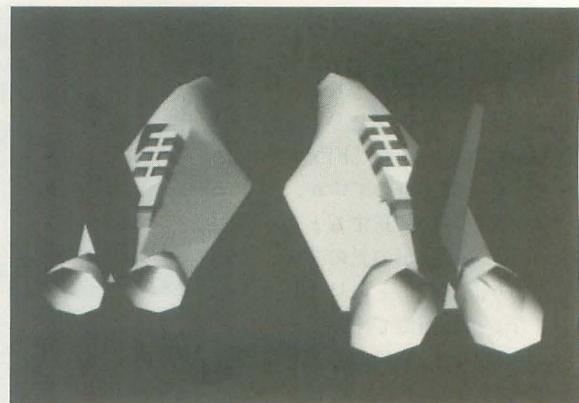



写真８ BOXTRANSで変形させたもの(右)

ということで，専門課程は特に大きな問題もなく，卒業することができます。まだまだCGAシステムは奥が深いとはいえ，モデリングからモーションデザイン，レンダリング，アニメーションまでひととおりはできるようになったはずです。おめでとうございます。

しかし，ここで第2の挫折が待っています。「ひととおりCGAはできるようになったはいいが，さて私は何を作ったらいいのだろうか？」つまり，作るべきCGAを見つけられないという問題です。

そりゃ，いきなりCGAコンテストで入賞できるような作品が作れれば，そんな問題で悩む必要はありませんが，やっと専門課程を卒業できた程度のレベルでは，GENIEに毛が生えた程度のCGAしかできません。そんな大きな目標をいきなり実行しようとすると，それこそ挫折を招くだけです。

CGAコンテストの入賞は，あくまで最終目標として，しばらくは習作やデモなどをいろいろ制作して，修業を積む必要があります。その間に，自分の好きなジャンル，得意なジャンルを見つけ，コンテストにエントリーする作品のアイデアを練りましょう。

それでは，ひととおりCGAシステムの使い方がわかったら，どのようなCGAを作るべきか，どのような道を進むべきか，考えてみましょう。もちろんその回答は1つではありません。以下のいくつかのアプローチのなかから，ご自分の興味のある方法を選んで実行すればよいと思います。

1） CGA大学院に進学する

もっとも一般的な道といえるでしょう。CGA大学には大学院があり，専門課程を卒業しても，その先には修士課程，博士課程があります。教養課程や専門課程は手取り足取りていねいに解説してくれていましたが，大学院では細かな説明は省略され，内容もだんだん難しくなってきます。最後の「人体型モデル理論研究」などは，理解できる人は全国に10人といないだろうといわれています。ですから，全部を完璧に理解する必要などありません。そのへんは気楽に考えてください。

大学院に進学するにあたって大切なのは，ただ読むのではなく，各自の自主的な研究を積極的に行うことです。「アトリビュートをこのように設定すると，こんな効果がある」とか，「スプライン曲線にはこのような特徴がある」と書いてあれば，ちゃんと実際にやってみてください。そして，それだけに留まらず，「では，こうしたらどうなるのだろう？」とか「こんなことが可能ではないか？」という疑問を持ち，試行錯誤してください。

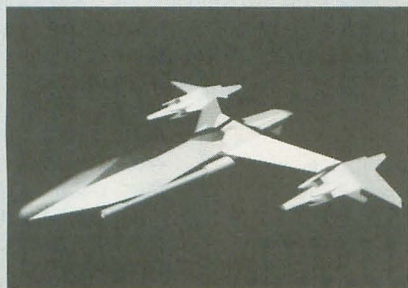
2） いろいろなツールを使ってみる

マニュアルの機能一覧編を見ながら，面白そうなツールをかたっぱしから試してみるというアプローチもあります。最初はマニュアルの例を試して，とりあえず使い方を把握し，あとはやはり積極的な試行錯誤が大切です。

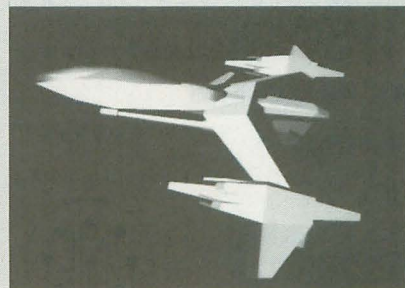
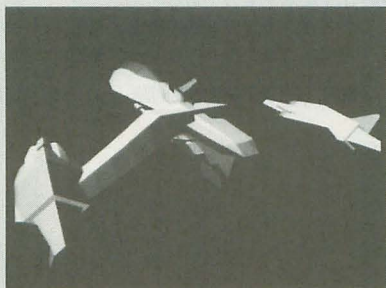
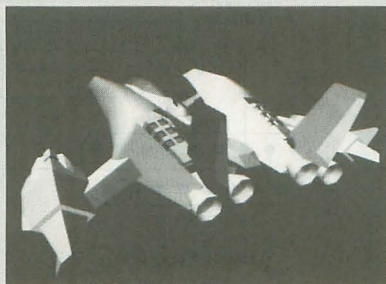
たとえば，前回使用したOUTLINEのマニュアルにはイルカも作れると書いてありますが，本当にそんなことができるのか試してみてはいかがでしょう。また，今回使用したBOXTRANSでは，変形していくアニメーションも可能と書いてあります。では，ディズニーアニメのように，車が急ブレーキをかけて，上半身が前にグニャリと曲がるような表現はできないものでしょうか？

さらに複数のツールを組み合わせることで，まったく別の表現も生まれてきます。アイデア次第で，予想もつかないCGAができるでしょう。

3） 過去のOh!Xを試す

この連載が始まってから，5年以上が経過しています。内容の有無はともかくとして，いろんな例を紹介してきました。「TORNADO」の1カットをマネしようとか，EPA2で爆発シーンを作ろうとか，実写の顔をマッピン

もとのX01号

ブースターを加えたX01号

写真9　X01号の修正結果

グしようとか……。そういった過去の連載を読み直し，実際自分でやってみてはいかがでしょうか？

なかには，内容が古くなっているものもあるでしょうが，考え方などは参考になると思います。また，最新のツールを使うことで，ずっと簡単にできるものもあるでしょう。

4） 映画の1シーンを作る

どんなカットを作るのか思いつかなければ，すでにある映画の1シーンを作るというのもよいアプローチです。カット割りやモーションデザインの勉強にもなります。

アマチュアCGA学会　論文集2

アマチュアCGA学会とは，高度な知識や，高価な機材を必要としない，それでいて実用性のあるCGA技術を発表する場である。

＊　　＊　　＊

「複数のX680x0によるフルカラーアニメーションに関する研究」

DōGA CGA研究所　高津　正道

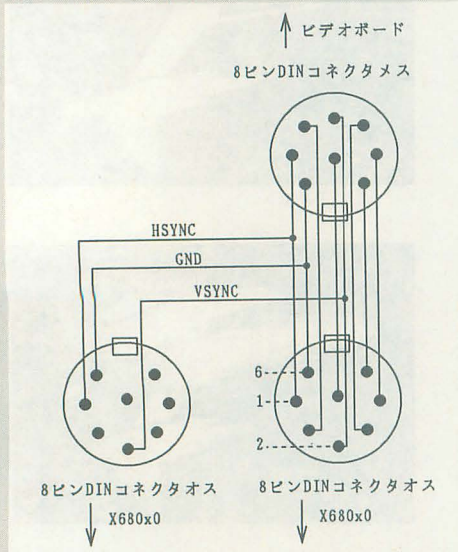
内容梗概

本稿では，X680x0を用いてフルカラー表示をするシステムを提案する。このシステムは複数のX680x0の画面表示を同期させることにより色数を増やすものであり，台数を増やすことで容易に色数，解像度を増やすことが可能になる。本システムはX680x0を用いたCGAの表現力を高めるのに有効であると思われる。

1．序論

パソコンの表示能力は年々進歩しており，現在は1600万色のフルカラーを表示するパソコンも珍しくない。そこで，本稿ではX680x0を用いてフルカラー表示をするシステム「DōGA-1 System S」を提案する。

このシステムは，まず，2台のX680x0でフルカラー画像の上位ビット部分，下位ビット部分を別々に表示させる。そして，2台のX680x0の画面表示を同期させたうえで，スーパーインポーズ機能を使って画面出力の信号を合成することでフルカラーの表示，アニメーションを実現するものである。

使用する2台のX680x0は同等である必要はない。通常，画像の上位ビットは圧縮効率がいいので，遅いマシンを使い，圧縮効率の悪い画像の下位ビット側に速いマシンを割り当てればよい。つまり，初代機と買い足したX68030を所有しているユーザーなどに特に適していると考えられる。

また本システムの応用として，色数を増やすのではなく，解像度を増やすことも可能であると思われる。その場合，2台のX680x0で画像の上位ビットと下位ビットを分担するのではなく，画面の上半分と下半分を分担すればよい。つまり，片方のX680x0で上半分の画面の512×256のアニメーションを，もう片方で下半分の512×256のアニメーションを行えば，合成された結果は512×512のアニメーションとなる。

さらに，理論的には，8台のX680x0を使うことにより，512×512ドット，1600万色，秒60コマのアニメーションも可能になることが予想される。

2．動作原理

2.1 画面表示の同期　X680x0は，15kHzモードではスーパーインポーズが可能である。このとき，X680x0はテレビコントロール端子から入力した同期信号にタイミングを合わせて画面表示を行っている。そこで，2台のX680x0の同期信号入力に同一の同期信号を入力すると，2台のX680x0はまったく同一のタイミングで画面表示を行うようになる。これを実現するための回路が図1に示すものである。

このとき，2台のX680x0の出力を何らかの方法を用いて合成すると，X680x0の画面にX680x0の画面をスーパーインポーズした画面を得ることができる。これを用いてフルカラー表示を実現する方法を2.2で，高解像度表示を実現する方法を2.3で述べる。

2.2 フルカラー表示システム
「DōGA-1 System S」

X680x0では，RGBそれぞれ0～31の，32段階の階調表現が可能であり，その結果，32×32×32＝32768色の表現が可能になっている。そこで，2台のX680x0の出力In1，In2に対し，合成出力をOut＝In1×32＋In2とすれば，このOutは0（＝0×32＋0）～1023（＝31×32＋31）の1024通りの値を取り得る。いい換えると，こうすることで出力側では1024段階の階調表現が可能になっているわけである。これをRGBそれぞれについて行えば，1024×1024×1024＝10億色の表現が可能になる。

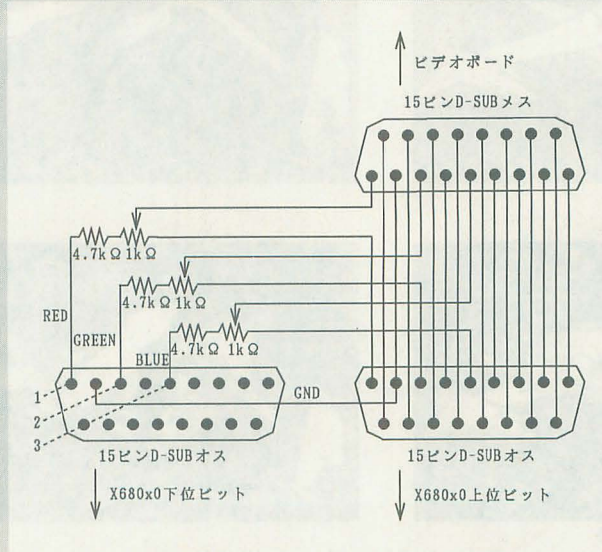
ただし，入出力する電圧は，もともと0.0～0.7Vなので，32倍してさらに片方の分を加えると，0.7Vを完全にオーバーしてしまう。そこで，合成した出力は，Out＝（In1×32＋In2）/33とする。これを実現する回路は図2のとおりである。

2.3 高解像度表示システム
「DōGA-1 System T」

「System S」では表示すべき画像を上位ビット，下位ビットに分割することでフルカラーを実現した。その代わりに，画像の上半分，下半分と分担すれば，表現できる解像度を倍に増やすことができる。厳密にいえば，もともと512×512の解像度で静止画の表示は可能であるが，その解像度ではなめらかなアニメーションを実行できなかったのを，512×256の画面2つに分割することにより，実用的な速度でのアニメーションを実現した。

ただし，この「System T」は理論上の存在であり，実際に開発，確認したわけではない。とりあえずアイデアの発表だけに留めておく。

図1　同期信号の分配回路

図2　RGB信号合成回路

3．実験結果

3.1 フルカラー表示システム「DōGA-1 System S」の作成

図1，図2の回路を組めば「DōGA-1」が完成する。厳密にいえば，図2での出力にはバッファアンプをつける必要がある。しかし，本システムのターゲットはCGAシステムを用い

「STAR WARS」などはよい題材でしょう。「TRON」などの初期のCG映画は，現在ではパソコンでもほぼ同等の映像を作ることも不可能ではありません。また，映画でなくても「ギャラクシアン[3]」や「スターブレード」といったゲームの１シーンを作ってもよいでしょう。

ただ，それらは著作権上おおっぴらに公開できないのが残念です。その場合，オリジナルのカットを加えたり，登場するメカをすべてGENIEで制作したものに差し替えるなどすれば，一応，問題はなくなると思います。

5) GENIEを発展させる

たアニメーションのビデオ出力であり，ビデオボードの使用を前提としている。ビデオボード内部にはバッファアンプが内蔵されているので，この回路ではバッファアンプを省略しても本システムは正常に動作し，システムの低価格化に貢献している。

本システムを使用したアニメーションビデオシステムのブロック図を図３に示す。

3.2 調整

「DōGA-1」を正常に動作させるためには，２台のX680x0の出力電圧を正確に32：１に合成する必要がある。この微妙な調整のためのプログラムをリスト１に示す。

上位ビット側で

dogaltest upper

を，下位ビット側で

dogaltest lower

を実行し，表示がなめらかなグラデーションになるように半固定抵抗を調整する。

3.3 アニメーション

DōGA-1を使ってフルカラーアニメーションを表示させるには，CGAシステム2.Zが必要である。まず，アニメーションさせたい画像を，RENDに-D1600オプションをつけて，1600万色フルカラーで作画させる。次に，できた画像を上位ビットと下位ビットに分割する。これは，Eights2Fives.r を使用する。

Eights2Fives ＜入力画像ファイル名＞ -o＜上位ファイル名＞ -w＜下位ファイル名＞

最後に，片方のX680x0で上位ビット画像を，もう片方のX680x0で下位ビット画像を，従来と同じように

HANIM -M2

でアニメーションさせれば，出力はフルカラーアニメーションとなる。このとき，よりマシンパワーのあるX680x0を下位ビット側に使ったほうがよい。

表示結果は，14ページのGraphic Galleryに示す。

4. 結論

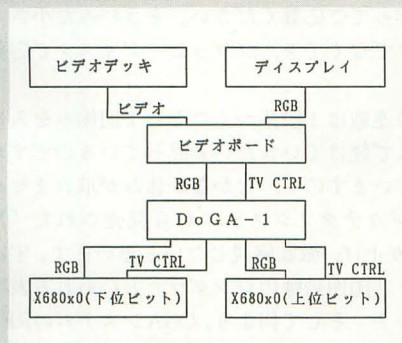
２台のX680x0を使ってフルカラー表示を行うシステム「DōGA-1」の提案を行った。また，このシステムは色数の向上だけでなく解像度の向上も可能であり，さらにはX680x0の台数を増やすことで表現能力の大幅な増加が可能である。

本システムは，CGAシステムを使ったビデオ作品の画質向上に役立つであろう。

図3 ブロック図

謝辞

最後に，さまざまな面でご協力，ご援助いただきましたプロジェクトチームDōGAの関係諸氏に深く感謝いたします。

参考文献

『Inside X68000』『Outside X68000』『X68030 Inside/Out』桒野雅彦，ソフトバンク刊
『X68000取扱説明書』SHARP
メガディスプレイへの道，瀧　康史，Oh!X，1994年6月号，pp.42-51.

図１のコネクタ

図２のコネクタ

追記１：部品一覧

部品	数量
8ピンDINコネクタオス	2
8ピンDINコネクタメス	1
15ピンD-SUBコネクタオス	2
15ピンD-SUBコネクタメス	1
4.7kΩ抵抗	3
1.0kΩ半固定抵抗	3
ユニバーサル基板	1
配線材	適量

追記２：なお，本研究の成果は，CGAコンテスト会場において展示する予定である。

リスト１

```
/*
        gcc dogaltest.c -ldos -liocs -lfloatfnc
*/
#include <stdio.h>
#include <iocslib.h>
#include <doslib.h>

#define UPPER 0
#define LOWER 1

void main(int argc, char *argv[])
{
    unsigned short *gram = (unsigned short*)(0xC00000);
    int x, y;
    int r, g, b;    /* 0 ... 1023 */
    int upperlower = LOWER;
    int sp;

    if (argc < 2) {
        printf("dogaltest upper ... カラーバーの上位ビットを表示する¥n");
        printf("dogaltest lower ... カラーバーの下位ビットを表示する¥n");
        exit(1);
    }
    if (tolower(argv[1][0]) == 'u') {
        upperlower = UPPER;
    }
    CRTMOD(13);
    G_CLR_ON();
    dummy_label1:
    sp = SUPER(0);
    for (y = 0; y < 512; ++y) {
        for (x = 0; x < 512; ++x) {
            r = g = b = y*2;
            if ((x / (512/7)) & 0x1) b = 0;
            if ((x / (512/7)) & 0x2) r = 0;
            if ((x / (512/7)) & 0x4) g = 0;
            if (upperlower == UPPER) {
                *gram++ = ((g & 0x3e0) << 6)
                        | ((r & 0x3e0) << 1)
                        | ((b & 0x3e0) >> 4);
            } else {
                *gram++ = ((g & 0x01f) << 11)
                        | ((r & 0x01f) << 6)
                        | ((b & 0x01f) << 1);
            }
        }
    }
    SUPER(sp);
    getch();
}
```

GENIEを気に入っていただいた方は，できたカットに手を加えていくというかたちで制作していくのもよいでしょう。たとえば背景をつけるとか，EPA2で各種エフェクトをつけるという方法もあります。また，FFEで設定できる動きにも限度があるので，CGA大学の修士課程を勉強して，フレームソースが自由に記述できるようになれば，さらに表現力を広げることができます。

6）モデリングを追求する

アニメーションするだけがCGAではありません。CGAの制作自体はちょっとおいといて，いかに精巧にモデリングするかを追求するという道もあります。

ただ，こういった方は，ガンダムとかパトレイバーとか，どうも著作権上問題のあるメカを作るというダークサイドの道に進んでしまうのが難点です。

著作権上問題のない，もっと汎用性のあるものを作れば，そのデータ集を当チームを通じて全国のユーザーに配布し，そのユーザーがそのデータを使って作品を作り，最終的には間接的にCGAコンテストにエントリーすることも夢ではありません。自分の作ったデータがグランプリ作品のなかに使われているというのも，ちょっと気分がいいと思いませんか。

7）リアリティを追求する

CGの歴史は，リアリティ追求の歴史といえます。CGAシステムは，どちらかといえば量産を重視しており，表現にはいろいろ制限がありますが，その制限のなかでリアリティを追求するというのも，興味深いアプローチです。

実写と区別がつかないようなCGAを作るにはどうしたらよいのでしょうか。まず，なぜCGとバレてしまうのかから考えないといけません。単に面数が少ないからか，微妙な汚れがないからか，遠くまでピントが合っているからか。考えると夜も眠れません。がんばってください。

8）曲に合わせる

作りたいCGAのイメージがなかなか湧かないという方でも，曲を聴いて，その曲のイメージに合わせてCGAを作ればイメージが湧きやすいでしょう。その種の作品の場合，音楽とタイミングをとる必要がありますが，CGAは0.05秒まで正確に編集できますから心配ありません。

ただ，当然，曲の著作権上の問題がありますので，コンテストの応募や上映には制限を受ける可能性があります。まぁ，習作と割り切れば問題ないでしょう。

過去にも，たとえば京大マイコンクラブが，F1グランプリのオープニングの曲に合わせ，各メーカーのパソコンを紹介するといった作品を見せてくれました。アイデア次第で面白いものができるでしょう。

9）作る理由を作る

ただなんとなくCGAを作ろうというだけでは，ちょっと行き詰まっただけで挫折してしまいます。どうしても，これを作らないといけない，完成しないとならないんだという理由，目的をまず見つけましょう。

学園祭用の客引きデモを作るとか，会社のプレゼンなどに一部CGAを取り入れるとか，友人たちに「CGAコンテストにエントリーする」と公言するとか，多少無理があってもいいじゃないですか。理由が何であれ，締め切りがはっきりしていることが大切です。

次のステップ

以上のようなさまざまなアプローチのいずれかを実行したならば，その過程でたくさんの習作カットができると思います。それらはバラバラで，何の脈絡もないでしょうが，とりあえずそれらをつなげてみましょう。そして，TCHED.Xで編集してみましょう。どのようにつなげるかは，CGA大学の「映像理論概論」を参考にしてください。

フルショットばかりでつながらないというなら，アップなどを作る必要性も出てくるでしょう。その場合，一から作るのではなく，習作として制作したフレームソースの視点の位置だけ変更すれば十分です。

そうやって，複数のカットをつなげると，1カットでは想像できなかった迫力，雰囲気が出てきます。1カットから1シーンへ発展したわけです。うまく作ればオリジナルブランドのフライングロゴのように仕上げても面白いでしょう。

さらに，その1シーンを2～3集めて，その間にタイトルを入れ，最後にエンドマークをつければ，CMか予告編，あるいはデモテープ，プロモーションビデオの出来上がりです。ちょっとした小作品になるわけです。

そして最後には，たまたまできたカットを寄せ集めるのではなく，こんな作品にしたいからこういうカットを作らなければいけないと，必要なカットを作るようになります。こうなれば，もう立派なCGA作家です。

でたらめにカットを作り，それを集めてデモを作り，計画をもって作品を作る……すべてのCGA作家が通ってきた道といえるでしょう。

おわりに

いよいよCGAコンテストの締め切りが近づいてきました。12月31日(必着)です。最近のコンテストのレベルは高いので，CGAシステムを初めて手に入れたばかりの方はあと1カ月で入選するような作品を作るのは難しいと思います。しかし，昨年のイントロダクション(紹介)コーナーで，選外になった作品もできるだけ多く取り上げる予定ですので，臆することなく，とりあえず寄せ集めのデモを作ってご応募ください。そういった小さい作品は，ビデオでなくても，フロッピーディスクで応募できます。

さて，この連載は1段落つくごとに1回休みを入れるというペースで続けていきたいと思っているのですが，ネタが押していますので，なかなか休みが取れません。来月はイマジカテクノシステムから発売された「XL/Image」を取り上げ，徹底解説したいと思います。宇宙人森山さん曰く「国内最強のレンダラ」といわれる表現力はいかほどか？　そして何より，CGAシステムのRENDの代わりとして使用した場合の実用性などに重点をおいて，調査してみたいと思います。

(で)のショートプロぱーてぃ —— その63

帰ってきました!

Komura Satoshi 古村 聡

今月のプログラムはゲームが2本にツール(?)が1本。ツールはディスプレイ上に花を咲かせ，心をなごませてくれます。ところで(で)氏は病から無事に復活を遂げました。これからどんな活躍を見せてくれるのでしょうか?

illustration : T.Takahashi

どもー。不肖(で)，恥ずかしながら，病院から帰ってまいりました。幸いなことに腫瘍は良性，手術も順調，その後の経過も良好だったんですけど……食事がつらかった。そう，腫瘍のできた場所が口の中だったもんだから，手術後の食事がず～っと流動食。しかもすごい味なんですね，これが。そう，あれは煎った小麦粉とハチミツとキュウリと小魚を口の中でぐちゃぐちゃと30分かみ続けて吐き出したような感じ。とても人間の食べ物じゃないですぜ。それにこの流動食，「1年間これだけを飲んでいても栄養失調にならない」といわれるほど栄養が凝縮されているそうです。ただ，それに体が対応できないと，腹がピーピーになってしまうというなかなか怖いもので，和式の白い便器にまたがりながら「早く人間になりたーい」と思っていた日々が……。結局はその流動食をかなり残して持ち帰ってきてしまいました。

しかし，そんな苦難の日々から脱出する日がきました。そう，ついに来週からはなにを食べてもいいし，コーヒーや酒を飲んでもOKという許可が出ましたんですー! 拍手～。よーっし，アンミラでパイとコーヒー飲み放題(ってことはミニスカ観察し放題)，そのあと……(中略)。

読者の皆様，また編集さんにご心配おかけして本当にもうしわけありませんでした。特に私の亡きあと乗っ取り……もとい，ショートプロの代打を務めてくれたてるてる君，サンクスでした。チミには余った流動食を全部プレゼントいたしましょう。栄養つけろよ。

今後ともどうぞ，(で)のショートプロぱーてぃをよろしくお願いいたします!

あなたOh!Xですかぁ?

まずは今月の1本目のプログラムにいきましょう。SX-BASIC用のゲームプログラムです。どうぞ!

GAMEOHX.SXB for SX-BASIC
(要SX-WINDOW ver.3.1,
SX-BASICこいのぼりバージョン)
埼玉県 市川賢二

まずはEDなどのエディタでリスト1を間違いなく入力(もみじ狩り以降のSX-BASICを使う場合は1行目の「)」のあとに「,0」を追加)して，ファイルを保存してください。次にSX-WINDOWを起動して，SXBASIC.Xのアイコンをクリックして SX-BASICを起動します。あとは先ほど保存したファイルをドラッグしてSX-BASICのウィンドウに放り込むだけです。

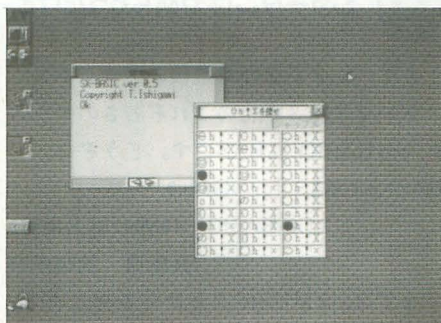
エラーが出ないで，無事動いたら遊び方。ゲームウィンドウ上に30個の「Oh!X」の文字が表示されます。でも，実は1個を除いて「◎h!X」(Oが二重丸になっている)とか「Oh!x」(Xが小文字になっている)など，どこかが間違って表示されています。正しい本物の「Oh!X」を捜し出して，マウスでクリックしてください。正解すると軽快(?)な音と共に当たりと表示されます。「シャッフル」のボタンをクリックすると，文字がシャッフルされて，再びプレイできます。

うーん，私もSX-BASICでリアルタイムゲームを作ってみたいんですけどね。でも，SX-BASIC上で，どうやって作ればいいのかわからなくて……。だから，リアルタイム性がなくてボタンを押せばOKってタイプのゲームを考えていたんです。でも，なかなか思いつかないんですよ。私も電卓だけは簡単に作れたんだけどなー。なるほど，こういうタイプのゲームなら作れるわけですね。アイデアに感心感心。

しかし，アイデアを形にしている市川さんを見ているとわれながら情けない。なにか作りたいものがあったら，それをイチから調べて作るべきだし，そうしないと力もつかないんでしょうけどね～。うーん，困ったもんだ。

皆さん，努力してます?

枯れ○○に花を咲かせましょう

さて，続いては心を和ませる環境ソフト。

GAMEOHX.SXB

2DPLANT.BAS

枯木，じゃなくてX68000に花を咲かせましょう。2DPLANT.BASです。どうぞ！

2DPLANT.BAS for X680x0

(X-BASIC)

鹿児島県　大上幸宏

このプログラムにはふたつのモードがあります。ひとつは鉢植えモード，もうひとつが花壇モードです。

鉢植えモードでは，１本の花を６種類の薬を使って育てていきます。

まず，下段のアイコンの「鉢植え」をクリックし，上段の薬品アイコンをいろいろ組み合わせて花を作ります。それぞれの薬品の効力は画面下のメッセージ表示エリアに表示されます。薬品の設定が終わったら作画に入ります。マウスを右クリックし，次に左クリックしてください。しばらくすると花が表示されます。

６つの薬品の効力は，

背丈：多いほど花の背が高くなる(49段階)
葉の量：多いほど葉の量が増える(64段階)
葉の形：ほとんどランダム(64種類)
花の色：量により色が異なる。白から始まって黄色，赤，紫，青，水色，と滑らかに変化する（49種類）
花弁の形：ほとんどランダム(64種類)
花弁の量：多くなると花弁の数が増える。ただ24を超すと花びらが重なる(49段階)

と，以上のようになっております。ただし，相手は生き物ですんで，同じ薬を与えても必ず同じような花が育つとは限らないですけどね(実はちょっとほかにもランダム要素があるのです)。

さて，もうひとつの花壇モードですが，花壇モードを選び，画面下のアイコン，「オート花壇」をクリックするとX68000が全自動でいろいろな花を100本作っていきます。100本の花をすべて描き終えるまでにかかる時間は，10MHzのX68000で，コンパイルしなければ，２時間ほどかかります。

また，このプログラムにはロード，セーブなどの機能もあります。

・「読み込み」アイコンで保存していた画像データを読み込みます。ファイル名を指定するときには，拡張子は指定しないでください。

・「保存」アイコンで，現在表示されている画面を拡張子GM3で保存します。このときファイル名には拡張子をつけないでください。

・「終了」で作業を終了します。キャンセルは「ESC」キーを押してください。

・「設定」でマウスの待ち時間の設定ができます。たぶん普段は使わないと思いますけどね。

以上で説明終わり。もう遊び方はパーフェクトにわかりましたよね？　これを作ったのは，もう常連の大上さんです。

私はこれ，気に入りました。薬品を使ってどんな花ができるかも楽しみだし，なんといっても，花壇モードにするといろんな花がでてきてきれい！　夢がありますよね，うんうん。

どうでもいいけど，私が入院してるときの見舞いの品にはなぜか花がなかったな〜。もらったものは，クッキー，ケーキ，ぎょうざ，タコヤキ，メロン，焼肉……なんか恨みでもあるんすか？　流動食しか食えないってのに……。そういえば，肌にすりこむ粗塩くれた人もいたけど風呂に入れないのにどうしろっちゅうんぢゃい。

関係ない話はこっちにおいといて，なになに？　投稿原稿によると「このプログラムにはちょっと残念なこともあります。それはなぜかコンパイルして使えないこと。このプログラム，コンパイラを使って.xファイルにして実行すると，なぜかハングアップしてしまうんです。うーん，なぜだっ？どなたかわかったら修正方法を教えてくれませんか？」とのこと。それでは，ってことで私が修正したものをリスト２に載せておきました。X-BASICって作り方を注意しないとコンパイルできないプログラムになっちゃうんですよね。そのへんの話が「ぷろぐらむ風まかせ」のほうに書いてあるので，X-BASICer(？)の皆さん，ぜひ読んでくださいね。

ジョーダンはやめちゃったけど

さて，いよいよ今月最後のプログラムでございます。３本目のこれぞ若さのゲーム(？)のBASKET.BASです。どうぞっ。

BASKET.BAS for X680x0

(X-BASIC　16MHz以上推奨)

兵庫県　浪越孝宏

まずは，ZMUSIC.Xを常駐させたあと，MUSICZ.FNCを組み込んだX-BASIC上で，リスト３のBASKET.BASをRUNしてくださいませ。そのとき「Z-MUSICシステムver.2.0」に収録されているRAP1.PCM(ACCENT)，THUD.PCM (EFFECTS)，APL.PCM (ADDITION)をカレントのディレクトリにコピーしておくか，環境変数zmusicにこれらのファイルのあるディレクトリをコマンドラインなどから，

set zmusic=A:¥ZMUSIC¥EFFECTS

というふうに設定しておいてください。PCMファイルはシュートをしたときの音，バウンドしたときの音，ゴールしたときの歓声になっております。できたかな？

BASKET.BAS

さて，それでは遊び方を説明いたしましょう。このゲームは，ショートプロでは割とよくあるタイプのバスケットのシュートをするゲームでキャラクターはテンキーで操作します。

キャラクター(仮にバークレー君とでもしとくか。あ，ハゲてないや(笑))は０キーで走り始め，１キーでジャンプ，２キーでシュートを打つ角度(強さ)が変わります。ジャンプしている間に２キーで素早くシュート角度を決めてシュートしましょう。でないと，バークレー君が地面に降りてしまいますよ。なんとなく，ショートプロって感じがしてきたでしょう。

シュートは入ってもはずれても，そのあとスペースキーを押すとゲーム終了，そのほかのキーで再びゲームスタートになります。また，ゴールの高さが毎回変わって，２回目からはシュートの成功率が表示されます。

む〜，リストが短いのにキャラクターがスプライトで表示されてるぞ〜。その上，ゴールのリングにボールが当たったときがリアルだし。ちゃんと本物のバスケのシュートみたいに，ボールがボワンボワンとリングやボードに跳ねて入るの。

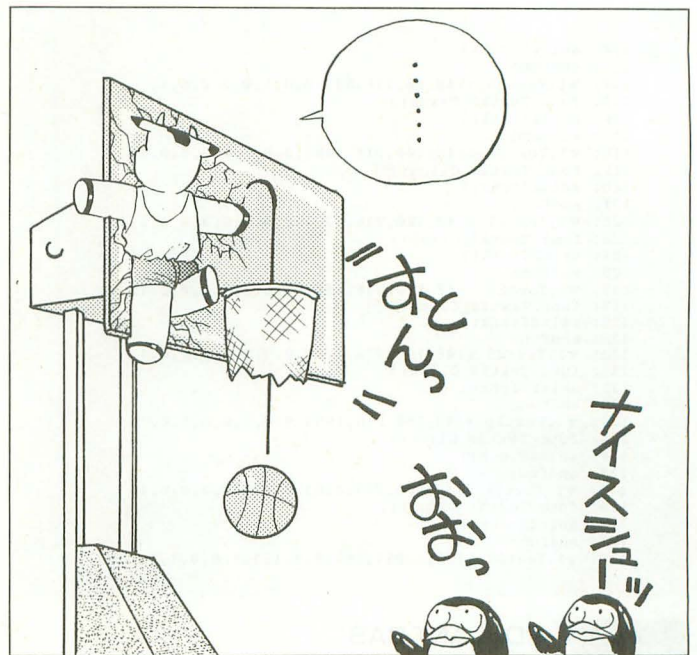
このプログラムの作者の浪越さんは「こ

の前，１時間ぐらいで作った小さなプログラムをショートプロ宛に気楽に送ったら，超豪華(？)な記念品が送られてきて恐縮してしまったので今度はがんばって作りました」ということであります。今度はがんばったカイあって掲載ですよ～。人間，まじめにやればいいことがあるもんですね。

で，このゲームなんですけど，実際にバスケットのシュートをするのと違って，遠くからジャンプ(ほとんどエアウォーク？)してレイアップのごとく，ぐぅ～っとほぼ真上に放ってしまうのが入りやすいですね。なんか，ジョーダンみたい。いや，私，バスケの選手ってバークレーとジョーダンしか知らないもんで。あ，あと岡山知ってる(苦笑)。

ゲームは一応16MHzのX68000でちょうどいいくらいに作ってあるようですが，10MHzでもなんとか遊べないことはないでしょう。全体に速度がゆっくりになりますけど，かえってスローモーションのような感じでおもしろいかもしれません。

そうそう，コンパイルするときや16MHz以上のマシンでスピードが速すぎるときはウエイトを調節してください。700行にウエイトを調整している部分があります。また，コンパイルして.xファイルで実行するときは，ZMUSIC.Xを常駐させるときにワークを大きめにとってくださいね。

さーてっと，今月はこれにて打ち止め。復活第１号の原稿も無事あがったし，焼肉でも食べにいこうかな～。やっぱり固形物が食べられるっていいよね。

それでは，幸せを噛みしめつつまた来月。

リスト1　GAMEOHX.SXB

```
  1: ▼Window Size (220,220),0,0,○h!Xを探せ
  2: str     d[8](30)=("○h!X","○h!X","○h!X",
  3:                   "●h!X","◎h!X","○h!X",
  4:                   "Oh!X","oh!X","のh!X",
  5:                   "Oh!X","Θh!X","Oh!X",
  6:                   "○h!×","○h!×","○h!×",
  7:                   "●h!×","◎h!×","○h!×",
  8:                   "Oh!×","oh!×","のh!×",
  9:                   "Oh!×","Θh!×","Oh!×",
 10:                   "○h!X","○h!X","○h!X",
 11:                   "●h!X","◎h!X","○h!X")
 12: int     m(30),i,sel=-1
 13: for i=0 to 29:m(i)=i:next
 14: Text32.caption="Wait!!"
 15: sha()
 16: Text32.caption="シャッフル"
 17: func File_Drop(filename;str)
 18: endfunc
 19: ▼1,Text1 (4,0,52,18),0,0,1,1,5,0,0,1,
 20: func Text1_Click()
 21: endfunc
 22: ▼1,Text2 (4,20,72,38),0,0,1,0,3,0,0,1,
 23: func Text2_Click()
 24: sel=0:chk()
 25: endfunc
 26: ▼1,Text3 (4,40,72,58),0,0,1,0,3,0,0,1,
 27: func Text3_Click()
 28: sel=1:chk()
 29: endfunc
 30: ▼1,Text4 (4,60,72,78),0,0,1,0,3,0,0,1,
 31: func Text4_Click()
 32: sel=2:chk()
 33: endfunc
 34: ▼1,Text5 (4,80,72,98),0,0,1,0,3,0,0,1,
 35: func Text5_Click()
 36: sel=3:chk()
 37: endfunc
 38: ▼1,Text6 (4,100,72,118),0,0,1,0,3,0,0,1,
 39: func Text6_Click()
 40: sel=4:chk()
 41: endfunc
 42: ▼1,Text7 (4,120,72,138),0,0,1,0,3,0,0,1,
 43: func Text7_Click()
 44: sel=5:chk()
 45: endfunc
 46: ▼1,Text8 (4,140,72,158),0,0,1,0,3,0,0,1,
 47: func Text8_Click()
 48: sel=6:chk()
 49: endfunc
 50: ▼1,Text9 (4,160,72,178),0,0,1,0,3,0,0,1,
 51: func Text9_Click()
 52: sel=7:chk()
 53: endfunc
 54: ▼1,Text10 (4,180,72,198),0,0,1,0,3,0,0,1,
 55: func Text10_Click()
 56: sel=8:chk()
 57: endfunc
 58: ▼1,Text11 (4,200,72,218),0,0,1,0,3,0,0,1,
 59: func Text11_Click()
 60: sel=9:chk()
 61: endfunc
 62: ▼1,Text12 (76,20,144,38),0,0,1,0,3,0,0,1,
 63: func Text12_Click()
 64: sel=10:chk()
 65: endfunc
 66: ▼1,Text13 (76,40,144,58),0,0,1,0,3,0,0,1,
 67: func Text13_Click()
 68: sel=11:chk()
 69: endfunc
 70: ▼1,Text14 (76,60,144,78),0,0,1,0,3,0,0,1,
 71: func Text14_Click()
 72: sel=12:chk()
 73: endfunc
 74: ▼1,Text15 (76,80,144,98),0,0,1,0,3,0,0,1,
 75: func Text15_Click()
 76: sel=13:chk()
 77: endfunc
 78: ▼1,Text16 (76,100,144,118),0,0,1,0,3,0,0,1,
 79: func Text16_Click()
 80: sel=14:chk()
 81: endfunc
 82: ▼1,Text17 (76,120,144,138),0,0,1,0,3,0,0,1,
 83: func Text17_Click()
 84: sel=15:chk()
 85: endfunc
 86: ▼1,Text18 (76,140,144,158),0,0,1,0,3,0,0,1,
 87: func Text18_Click()
 88: sel=16:chk()
 89: endfunc
 90: ▼1,Text19 (76,160,144,178),0,0,1,0,3,0,0,1,
 91: func Text19_Click()
 92: sel=17:chk()
 93: endfunc
 94: ▼1,Text20 (76,180,144,198),0,0,1,0,3,0,0,1,
 95: func Text20_Click()
 96: sel=18:chk()
 97: endfunc
 98: ▼1,Text21 (76,200,144,218),0,0,1,0,3,0,0,1,
 99: func Text21_Click()
100: sel=19:chk()
101: endfunc
102: ▼1,Text22 (148,20,216,38),0,0,1,0,3,0,0,1,
103: func Text22_Click()
104: sel=20:chk()
105: endfunc
106: ▼1,Text23 (148,40,216,58),0,0,1,0,3,0,0,1,
107: func Text23_Click()
108: sel=21:chk()
109: endfunc
110: ▼1,Text24 (148,60,216,78),0,0,1,0,3,0,0,1,
111: func Text24_Click()
```

```
112: sel=22:chk()
113: endfunc
114: ▼1,Text25 (148,80,216,98),0,0,1,0,3,0,0,1,
115: func Text25_Click()
116: sel=23:chk()
117: endfunc
118: ▼1,Text26 (148,100,216,118),0,0,1,0,3,0,0,1,
119: func Text26_Click()
120: sel=24:chk()
121: endfunc
122: ▼1,Text27 (148,120,216,138),0,0,1,0,3,0,0,1,
123: func Text27_Click()
124: sel=25:chk()
125: endfunc
126: ▼1,Text28 (148,140,216,158),0,0,1,0,3,0,0,1,
127: func Text28_Click()
128: sel=26:chk()
129: endfunc
130: ▼1,Text29 (148,160,216,178),0,0,1,0,3,0,0,1,
131: func Text29_Click()
132: sel=27:chk()
133: endfunc
134: ▼1,Text30 (148,180,216,198),0,0,1,0,3,0,0,1,
135: func Text30_Click()
136: sel=28:chk()
137: endfunc
138: ▼1,Text31 (148,200,216,218),0,0,1,0,3,0,0,1,
139: func Text31_Click()
140: sel=29:chk()
141: endfunc
142: ▼1,Text32 (132,1,214,19),0,0,1,1,7,0,0,1,シャッフル
143: func Text32_Click()
144: Text32.caption="Wait!!"
145: sha()
146: Text32.caption="シャッフル"
147: Text1.caption=" "
148: endfunc
149: func chk()
150: if m(sel)<>0 then Text1.caption="はずれ"
151: if m(sel)=0 then beep:Text1.caption="当たり!"
152: endfunc
153: func sha()
154: int w,n1,n2,i
155: for i=0 to 100
156:   n1=rnd()*30:n2=rnd()*30:w=m(n1):m(n1)=m(n2):m(n2)=w
157: next
158: Text2.caption=d(m(0)):      Text3.caption=d(m(1))
159: Text4.caption=d(m(2)):      Text5.caption=d(m(3))
160: Text6.caption=d(m(4)):      Text7.caption=d(m(5))
161: Text8.caption=d(m(6)):      Text9.caption=d(m(7))
162: Text10.caption=d(m(8)):     Text11.caption=d(m(9))
163: Text12.caption=d(m(10)):    Text13.caption=d(m(11))
164: Text14.caption=d(m(12)):    Text15.caption=d(m(13))
165: Text16.caption=d(m(14)):    Text17.caption=d(m(15))
166: Text18.caption=d(m(16)):    Text19.caption=d(m(17))
167: Text20.caption=d(m(18)):    Text21.caption=d(m(19))
168: Text22.caption=d(m(20)):    Text23.caption=d(m(21))
169: Text24.caption=d(m(22)):    Text25.caption=d(m(23))
170: Text26.caption=d(m(24)):    Text27.caption=d(m(25))
171: Text28.caption=d(m(26)):    Text29.caption=d(m(27))
172: Text30.caption=d(m(28)):    Text31.caption=d(m(29))
173: endfunc
```

リスト2 2DPLANT.BAS

```
  10 /*--------------------------------------
  20 /*-          2D PLANTA                 -
  30 /*-  copyright by Y.Oue 1994 PX9402B   -
  40 /*--------------------------------------
  50 screen 1,2,1,1:console 0,32,0
  60 int a,b,c,d,x,y,i,j,fi,fj,fk,hi,hj,hk,ji
  70 int o1,o2,o3,o4,o5,o6,o7,oc1,oc2,ha,pp
  80 int max,may,mix,miy,ux,dx,dy,lb,rb,kx,vv
  90 int ha1,hhp1,hpd1,cr,d1,d2,x1,y1,er,fp,wa
 100 float rd,co,si,sa,va,sa1,va1,ha2
 110 char f1,f2,f3
 120 str hp1[255],hp2[255],fn[255],in
 130 dim str ld(20)[255]
 140 dim int lx(1,10,11),ly(1,10,11),gx(10),gy(10)
 150 dim int r1(11),r2(11),lz(1,10,11),bd(5,15,15)
 160 dim int ls(3),cf(3),fsx(15),fsy(15),rr(2)
 170 dim int mx(11),my(11),mz(11),ldx(99),ldy(99)
 180 dim char hhd(6),hhh(2),hpd(63,2),hsp(6),rc(2),rc1(2),cmd(5)
 190 dim float cx(11),cy(11),px(10),py(10),gg(2048),hv(7),hh(3)
 200 dim str fs(6)[255],syb(11),mu(5),mg(5),hhp(6,3,1)[30],fhp(6
,1)[30]
 210 ls={65278,65278,61423,61423}:cf={1,2,4,5}
 220 fsx={1,1,1,0,-1,-1,-1,0,2,0,-2,0,3,0,-3,0}
 230 fsy={1,0,-1,-1,-1,0,1,1,0,-2,0,2,0,-3,0,3}
 240 syb={"L","O","S","A","Q","U","A","D","V","E","I","T"}
 250 mu={"鉢植え","オート花壇","読み込み","保存","終了","設定"}
 260 mg={"背丈","葉の量","葉の形","花の色","花弁の形","花弁の量"}
 270 hsp={0,3,2,1,1,0,0}:hv={0.78#,0.75#,0.62#}:hh={0,1.7#,2,2.5
#}
 280 cmd={48,63,63,48,63,48}:wa=1
 290 vpage(0):apage(0):mouse(4):mouse(1)
 300 INIT():vpage(1)
 310 randomize(val(right$(time$,2))+(val(mid$(time$,4,2))*100))
 320 /** main **
 330 COM()
 340 end
 350 func COM():/** コマンド受け付け **
 360 msarea(70,418,196,439)
 370 repeat
 380   msstat(dx,dy,lb,rb):mspos(dx,dy)
 390   for i=0 to 5
 400     if (((i*21)+70)<dx) and (dx<((i*21)+91)) then ux=kx:kx=
i
 410   next
 420   if ux<>kx then palet(ux+36,42280)
 430   palet(kx+36,59192)
 440   locate 5,29:print "command: "+mu(kx);:print spc(25)
 450   if lb<>0 then {
 460     switch kx
 470       case 0:POD():break
 480       case 1:AUTOD():break
 490       case 2:F_LOAD():break
 500       case 3:F_SAVE():break
 510       case 4:PQUIT():break
 520       case 5:CONFIG()
 530     endswitch
 540     for i=36 to 40:palet(i,42280):next
 550   msarea(70,418,196,439) }
 560 until a=1
 570 endfunc
 580 func POD():/** 鉢植え **
 590 msarea(70,398,196,417):b=0:LOOP(2)
 600 repeat
 610   msstat(dx,dy,lb,rb):mspos(dx,dy)
 620   for i=0 to 5
 630     if (((i*21)+70)<dx) and (dx<((i*21)+91)) then ux=kx:kx=
i
 640   next
 650   if ux<>kx then palet(ux+30,42280)
 660   palet(kx+30,59192)
 670   locate 5,29:print mg(kx);:print spc(15)
 680   if lb<>0 then {
 690     if bd(kx,0,0)=cmd(kx) then bd(kx,0,0)=0 else bd(kx,0,0)
=bd(kx,0,0)+1
 700   GRP(kx,0,0):LP() }
 710   if rb<>0 then {
 720     locate 5,29:print"左クリック・・・作画   右クリック・・・終了"
 730     LOOP(2)/*コンパイルする場合，数値を大きくしてみてください**
 740     repeat:d=0
 750       msstat(dx,dy,lb,rb)
 760       if lb<>0 then d=2:MPD(0,0,0):f1=1
 770       if rb<>0 then d=1:b=1
 780     until d<>0
 790   locate 5,29:print spc(50) }
 800 until b=1
 810 for i=30 to 35:palet(i,42280):next
 820 endfunc
 830 func GRP(o1;int,o2;int,o3;int):/** グラフ **
 840 apage(0)
 850 if ((o1<>1) and (o1<>2) and (o1<>4)) then box(204,400+(o1*3
),204+(bd(o1,o2,o3)/2),401+(o1*3),20+o1)
 860 if (o1=1) or (o1=2) or (o1=4) then box(204,400+(o1*3),204+(
bd(o1,o2,o3)/3),401+(o1*3),20+o1)
 870 if bd(o1,o2,o3)=0 then box(204,400+(o1*3),254,401+(o1*3),14
)
 880 endfunc
 890 func LP():/** 数値表示 **
 900 for fi=0 to 1:for fj=0 to 2
 910     locate 31+7*fi,25+fj:print chr$(fi*3+fj+65);
 920     print using"  ##";bd(fi*3+fj,hi,hj)
 930 next:next
 940 endfunc
 950 func MPD(o1;int,o2;int,o4;int):/** 植物作成 **
 960 if (o4=0) or (f1=1) then apage(0):fill(100,70,400,370,14):f
1=0
 970 for fi=0 to 2:rr(fi)=int(rnd()*3):next
 980 d1=bd(2,o1,o2) ¥ 8:d2=bd(2,o1,o2) mod 8
 990 MOF(d1,d2,0):vv=1
1000 f2=int(rnd()*7)
1010 for fi=0 to 1
1020   for fj=0 to 2:hpd1=hpd(bd(1,o1,o2),fj)
1030     if hpd1<>0 then {
1040       if hpd1=5 or hpd1=6 then rr(fj)=0
1050       hhp1=val(left$(hhp(hpd1,rr(fj),fi),1))
1060       if hhp1<>0 then {
1070         for fk=1 to hhp1
1080           sa1=asc(mid$(hhp(hpd1,rr(fj),fi),((fk-1)*3)+2,1))
-48
1090           DDR1(o1,o2,sa1,int(1+fj*((bd(0,o1,o2)/5)+1)*1.3#)
,hv(fj),1.7#,2,3,o4)
1100         next }
1110     }
1120   next
1130   if fi=0 then {
1140     if o4=0 then kkx=236:kky=210
1150     if o4=1 then kkx=20*ldx((o1*10)+o2)+70+f2:kky=15*ldy((o
1*10)+o2)+60
1160     apage(0):fill(kkx+1,kky-(bd(0,o1,o2)/2*vv),kkx+1,kky+38
,2)
1170   box(kkx,kky-(bd(0,o1,o2)/2*vv),kkx+2,kky+38,3) }
1180 next
1190 if 27<bd(5,o1,o2) then cr=1 else cr=0
1200 y=(bd(0,o1,o2)/2)+33
```

▶大学院に受かったのでぼーとしてます。　　小畑　達哉(23)大阪府

```
1210 apage(0):fill(kkx,kky+31-y,kkx+3,kky+35-y,49)
1220 fill(kkx-1,kky+32-y,kkx+4,kky+34-y,49)
1230 line(kkx-2,kky+33-y,kkx+5,kky+33-y,49):apage(1)
1240 d1=bd(4,o1,o2) ¥ 8:d2=bd(4,o1,o2) mod 8
1250 MOF(d1,d2,1)
1260 for fj=0 to cr
1270   for fi=0 to 1:hpd1=(bd(5,o1,o2)/4)-(cr*7)
1280      if (cr=1) and (fj=0) then hpd1=6
1290     if hpd1=0 then hpd1=1
1300     hhp1=val(left$(fhp(hpd1,fi),1))
1310     if hhp1<>0 then {
1320       for fk=1 to hhp1
1330           sa1=asc(mid$(fhp(hpd1,fi),((fk-1)*3)+2,1))-48
1340           DDR1(o1,o2,sa1+(0.5#*fj),-1,0.9#-(fj*0.2#),1+(fj*
0.4#),bd(3,o1,o2)+100,bd(3,o1,o2)+150,o4)
1350     next }
1360   next
1370 next
1380 endfunc
1390 func DDR1(o1;int,o2;int,sa;float,ha;int,va;float,ha2;float,
oc1;int,oc2;int,o4;int):/** 画面描画 **
1400 max=0:may=0:mix=136:miy=136
1410 rd=sa*22.5#/(180/pi()):co=cos(rd):si=sin(rd)
1420 for i=0 to 11
1430   cx(i)=((mx(i))*co-(my(i))*si)*va
1440   cy(i)=((mx(i))*si+(my(i))*co)/2*va-(mz(i)+ha)*ha2
1450   if (cx(i)+63)>max then max=cx(i)+63
1460   if (cx(i)+63)<mix then mix=cx(i)+63
1470   if (cy(i)+103)>may then may=cy(i)+103
1480   if (cy(i)+103)<miy then miy=cy(i)+103
1490 next
1500 apage(1):fill(1,1,136,136,0)
1510 box(0,0,137,137,4):paint(1,1,oc1)
1520 if o4=0 then x1=175:y1=140
1530 if o4=1 then {
1540 x1=20*ldx((o1*10)+o2)+9+f2:y1=15*ldy((o1*10)+o2)-10 }
1550 for i=0 to 11
1560   line(cx(r1(i))+63,cy(r1(i))+103,cx(r2(i))+63,cy(r2(i))+10
3,oc2)
1570 next
1580 if ha=-1 then y=(bd(0,o1,o2)/2)+33 else y=0
1590 paint(1,1,0)
1600 for i=mix to max:for j=miy to may
1610     pp=point(i,j)
1620     if pp<>0 then apage(0):pset(i+x1,j+y1-y,pp):apage(1)
1630 next:next
1640 endfunc
1650 func MOF(o5,o6,o7):/** モーフィング **
1660 for fi=0 to 11
1670   mx(fi)=int((lx(o7,o5,fi)+lx(o7,o6,fi))/2)
1680   my(fi)=int((ly(o7,o5,fi)+ly(o7,o6,fi))/2)
1690   mz(fi)=int((lz(o7,o5,fi)+lz(o7,o6,fi))/2)
1700 next
1710 endfunc
1720 func AUTOD():/** オート花壇作り **
1730 locate 30,29:print"作成中"
1740 for hi=0 to 9
1750   for hj=0 to 9
1760     for hk=0 to 5
1770       bd(hk,hi,hj)=int(rnd()*cmd(hk))
1780     next
1790     locate 38,29:print using"###/100";(hi*10)+hj+1
1800     LP():MPD(hi,hj,1)
1810 next:next
1820 endfunc
1830 func F_LOAD():/** 画面ロード **
1840 locate 5,30:input"ファイル名(拡張子は省略)";fn
1850 locate 5,30:print"ロード中";:print spc(35)
1860 apage(0):error off:er=img_load(fn+".GM3",0,0,8):error on
1870 if er<>0 then locate 5,30:print"ファイルがありません";:print spc(
30):LOOP(10)
1880 locate 5,30:print spc(55)
1890 endfunc
1900 func F_SAVE():/** 画面セーブ **
1910 locate 5,30:input"ファイル名(拡張子は省略)";fn
1920 error off:fp=fopen(fn+".GM3","r"):error on
1930 if fp=-1 then locate 5,30:print"新規保存";:print spc(35)
1940 if fp<>-1 then { locate 5,30:print"上書きしますか(cancel[ESC])"
;:print spc(20)
1950   repeat:until inkey$(0)=""
1960   repeat:in=inkey$(0):fcloseall()
1970     if asc(in)=27 then locate 5,30:print spc(40):return()
1980   until in<>""
1990 locate 5,30:print"上書きします";:print spc(20) }
2000 apage(0):img_save(fn,0,0)
2010 locate 5,30:print spc(50)
2020 endfunc
2030 func PQUIT():/** 終了 **
2040 locate 5,30:print"終了します(cancel[ESC])"
2050 repeat:until inkey$(0)=""
2060 repeat:in=inkey$(0)
2070   if asc(in)=27 then locate 5,30:print spc(40):return()
2080 until in<>""
2090 cls:apage(0):wipe():apage(1):wipe():mouse(0):mouse(2):end
2100 endfunc
2110 func CONFIG():/** 設定 **
2120 locate 5,30:print"ウエイト設定 現在 ";wa
2130 locate 30,30:input wa
2140 locate 5,30:print spc(50)
2150 endfunc
2160 func LOOP(op)
2170 for ji=0 to op*500:next
2180 endfunc
2190 func INIT():/** 初期設定 **
2200 locate 0,0:print"ポイントデータ登録"
2210 /*      <     ポイントデータ     ><Z  データ>
2220  ld(0)="0012212<<3:IE?@JGDFKHHIL222334433221"
2230  ld(1)="0058756DC7COLCJOOHLNNJML011222211111"
2240  ld(2)="0013314984<CC=EKJDLOOLOO011223333221"
2250  ld(3)="00798789>8?<F?GLLFLNNKOO111222211110"
2260  ld(4)="008;:93@>33EC49I=9EIHEMM022222222111"
2270  ld(5)="00:;;::HH:@FF@MCF@DFFDOO122222222222"
2280  ld(6)="0056650<<07KK7:CC:GMMGOO022222222110"
2290  ld(7)="0008806BB6?KK?BMMBIOOIOO000112233444"
2300 ld(10)="44255337657;;9:==<=?>=??000111122222"
2310 ld(11)="44355348848==8=@@==>>=??011110000000"
2320 ld(12)="44466438834<<46>>68??8==022333333222"
2330 ld(13)="33244248846;;69??9<??<>>000001111111"
2340 ld(14)="331441166118812992599577000112233444"
2350 ld(15)="3314412;;26@@6;@@;=??===011222211111"
2360 ld(16)="4436638>>8=CC=:<<:688644011111111110"
2370 ld(17)="551::11<<11>>12@@27AA7==000112233445"
2380 for k=0 to 1
2390   for j=0 to 7
2400     for i=0 to 11
2410       lx(k,j,i)=asc(mid$(ld(j+k*10),(i*2)+1,1))-48
2420       ly(k,j,i)=asc(mid$(ld(j+k*10),(i*2)+2,1))-48
2430       lz(k,j,i)=asc(mid$(ld(j+k*10),25+i,1))-48
2440 next:next:next
2450 r1={0,0,1,2,3,4,5,6,7,8,9,10}
2460 r2={1,2,3,4,5,6,7,8,9,10,11,11}
2470 locate 30,0:print"Ok":print"葉発生パターンデータ登録"
2480 hp1="000100010200020110300030210400220310320410303033420330
403223033430322403500050440422332510432423520502052442433530503
0534435405225325425335445506006106205526026305536406226326426336
43650652653"
2490 for i=0 to 63
2500   for j=0 to 2
2510     hpd(i,j)=asc(mid$(hp1,(i*3)+j+1,1))-48
2520 next:next
2530 hhp(1,0,0)="0":hhp(1,0,1)="1011":/** 葉発生パターンデータ**
2540 hhp(1,1,0)="0":hhp(1,1,1)="1411"
2550 hhp(1,2,0)="1811":hhp(1,2,1)="0"
2560 hhp(1,3,0)="0":hhp(1,3,1)="1<11"
2570 hhp(2,0,0)="1811":hhp(2,0,1)="1011"
2580 hhp(2,1,0)="1:11":hhp(2,1,1)="1211"
2590 hhp(2,2,0)="1511":hhp(2,2,1)="1=11"
2600 hhp(2,3,0)="1611":hhp(2,3,1)="1>11"
2610 hhp(3,0,0)="1611":hhp(3,0,1)="2011<11"
2620 hhp(3,1,0)="1811":hhp(3,1,1)="2311=11"
2630 hhp(3,2,0)="2911511":hhp(3,2,1)="1?11"
2640 hhp(3,3,0)="1911":hhp(3,3,1)="2411>11"
2650 hhp(4,0,0)="1811":hhp(4,0,1)="3011411<11"
2660 hhp(4,1,0)="2611:11":hhp(4,1,1)="2211>11"
2670 hhp(4,2,0)="2711;11":hhp(4,2,1)="2311?11"
2680 hhp(4,3,0)="2511911":hhp(4,3,1)="2=11111"
2690 hhp(5,0,0)="3612811:12":hhp(5,0,1)="5011212411<11>12"
2700 hhp(6,0,0)="7502603711802903:11;02"
2710 hhp(6,0,1)="9<03=11>02?030111022203311402"
2720 fhp(1,0)="2611;11":fhp(1,1)="1111"
2730 fhp(2,0)="1811":fhp(2,1)="3011411<11"
2740 fhp(3,0)="2611911":fhp(3,1)="3011311<11"
2750 fhp(4,0)="3611;11811":fhp(4,1)="4>11111311"
2760 fhp(5,0)=hhp(5,0,0):fhp(5,1)=hhp(5,0,1)
2770 fhp(6,0)=hhp(6,0,0):fhp(6,1)=hhp(6,0,1)
2780 hp1="98:79;68:<579;=468:<>3579;=?2468:<>@13579;=?A02468:<>@
B13579;=?A2468:<>@3579;=?468:<>579;=68:<79;8:9"
2790 hp2="011222333344444555555666666777777788888888899999999
9::::::::;;;;;;;;<<<<<<<======>>>>>????@@@AAB"
2800 for i=0 to 99
2810   ldx(i)=asc(mid$(hp1,i+1,1))-48
2820   ldy(i)=asc(mid$(hp2,i+1,1))-48
2830 next
2840 locate 30,1:print"Ok":print"パレット初期化"
2850 for i=0 to 48
2860   rc(0)=32:rc(1)=32:rc(2)=32
2870   for fi=0 to i
2880     if fi>0 and fi<9 then rc(2)=rc(2)-4
2890     if fi>8 and fi<17 then rc(1)=rc(1)-4
2900     if fi>16 and fi<25 then rc(2)=rc(2)+4
2910     if fi>24 and fi<33 then rc(0)=rc(0)-4
2920     if fi>32 and fi<41 then rc(1)=rc(1)+4
2930     if fi>40 and fi<48 then rc(0)=rc(0)+4
2940   next
2950   if i=0 or i=48 then rc(0)=32:rc(1)=32:rc(2)=32
2960   for fi=0 to 2
2970     if rc(fi)=32 then rc(fi)=rc(fi)-1
2980     if rc(fi)>5 then rc1(fi)=rc(fi)-4 else rc1(fi)=rc(fi)
2990   next
3000   palet(i+100,rgb(rc(0),rc(1),rc(2))):palet(i+150,rgb(rc1(0
),rc1(1),rc1(2)))
3010 next
3020 palet(2,51464):palet(3,30918):palet(13,38052)
3030 palet(14,42280):palet(20,64148):palet(21,64788)
3040 palet(22,65492):palet(23,22484):palet(24,22526)
3050 palet(25,47614):palet(49,65472)
3060 locate 30,2:print"Ok":print"画面作成"
3070 for i=30 to 41:palet(i,42280):next
3080 for i=0 to 84:for j=0 to 3
3090 line(0,(6*i)+cf(j),511,(6*i)+cf(j),13,ls(j)):next:next
3100 fill(31,11,479,451,14)
3110 fill(31,460,479,503,14)
3120 fs(0)="<0815==2=801066656":fs(1)="91706;6;6;10100"
3130 fs(2)="C2315=23232070;1;07349>56":fs(4)="74715=949>82<"
3140 fs(3)="<3315929290?02232=":fs(5)="=57154=94:6?6?=2<22"
3150 for fi=0 to 2:for fj=0 to 2
3160 FONT(fi,fj,0,0):next:next
3170 FONT(1,1,19,1)
```

▶受験生はつらい……(誰でもそうだと思うけど)。よりによって試験中とか勉強中とかに「やりたい！」というコトが頭に浮かんでしまう。……マーキュリーボード作りたい。

河上　博仁(18)埼玉県

```
3180 symbol(366,438,"YUKISOFT",1,1,0,1,0)
3190 for i=0 to 1:for j=0 to 5
3200   fill(70+j*21,395+i*23,91+j*21,416+i*23,(i*6)+j+30)
3210 next:next
3220 locate 30,3:print"Ok"
3230 ICON():cls
3240 endfunc
3250 func FONT(o1;int,o2;int,o3;int,o4;int):/** ロゴ表示 **
3260 for i=0 to 5:o3=o3+o4
3270   x=val(mid$(fs(i),2,2))+413+o1
3280   y=val(mid$(fs(i),4,2))+431+o2
3290   for j=6 to asc(left$(fs(i),1))-42
3300     as=asc(mid$(fs(i),j,1))-48
3310     line(x,y,x+fsx(as),y+fsy(as),o3)
3320     x=x+fsx(as):y=y+fsy(as)
3330 next:next
3340 endfunc
3350 func ICON():/** アイコン表示 **
3360 box(73,421,88,425,1)
3370 line(75,425,76,436,1):line(76,436,85,436,1)
3380 line(85,436,86,425,1):paint(75,423,23):paint(82,428,23)
3390 line(92,429,101,436,1):line(101,436,110,429,1)
3400 line(110,429,101,422,1):line(101,422,92,429,1)
3410 paint(95,429,21)
3420 for i=0 to 5:for j=0 to 1
3430 symbol(113+i*11,418+j*10,syb(j*6+i),1,1,0,1,0):next:next
3440 box(184,421,187,436,1):paint(185,423,22)
3450 fill(181,423,190,426,1)
3460 for i=0 to 5
3470   box(76+i*21,398,85+i*21,401,1)
3480   line(77+i*21,401,77+i*21,403,1)
3490   line(84+i*21,401,84+i*21,403,1)
3500   line(77+i*21,403,74+i*21,406,1)
3510   line(84+i*21,403,87+i*21,406,1)
3520   line(74+i*21,406,74+i*21,414,1)
3530   line(87+i*21,406,87+i*21,414,1)
3540   line(75+i*21,415,86+i*21,415,1)
3550   line(74+i*21,406,86+i*21,406,1)
3560   paint(80+i*21,410,i+20)
3570 next
3580 endfunc
```

リスト3 BASKET.BAS

```
  10 /*******************************************************
  20 /*  THE BASKET                                  1994/6/18
  30 /*                           programmed by T.Namikoshi
  40 /*******************************************************
  50 screen 0,3,1,1:console ,,0:m_init():sp_init()
  60 sp_disp(1):sprite_pattern():sprite_pallet():sp_on()
  70 float g=40,bvx,bvy,mvy,winrate,xtest,ytest
  80 int bx,by,bxcnt,bycnt,mx,my,mxcnt,mycnt,mode
  90 int goal,angle,playcnt=0,wincnt=0,high=0
 100 dim int vector(2,9)={18,17,16,14,12,9,6,2,0,-7
 110    ,-7,-1,4,8,11,14,16,18,19,20
 120    ,200,400,600,800,1000,1200,1400,1600,1800,2000 }
 130 m_pcmset(0,"rap1.pcm"):m_pcmset(1,"thud.pcm")
 140 str getkey:m_pcmset(2,"apl.pcm")
 150 fill(3,210,255,255,900):box(2,0,255,255,900)
 160 j=0:for i=3 to 9:j=j+i:line(3,210+j,255,210+j,1):next
 170 circle(180,50,30,900,0,360,270)
 180 symbol(130,25,"THE BASKET",1,2,2,8872,0)
 190 symbol(129,24,"THE BASKET",1,2,2,15218,0)
 200 /********** main loop ***********************************
 210 while 1:goal=0:mode=0:angle=0:high=int(rnd()*50)+1
 220 bx=190:bvx=-1:bxcnt=0:mx=200:mxcnt=0:drawbg(high)
 230 by=206:bvy=-500:bycnt=0:my=200:mvy=0:mycnt=0
 240 while 1
 250   if bxcnt=1 then {
 260     bx=bx+bvx:if bx<0 then bvx=1
 270     if(bx>11)*(bx<23)*(by>(-12+high))*(by<(48+high))then{
 280       bvx=-bvx:bvy=bvy+200:m_pcmplay(1,3,4)
 290       if (by<(-9+high))+(by>(45+high)) then by=by-sgn(bvy)*
3:bvy=-bvy+200
 300     }
 310     xtest=bx-34.5#:ytest=by-32.5#-high
 320     if(abs(xtest)<5)*(abs(ytest)<5) then{m_pcmplay(1,3,4)
 330       bvx=sgn(xtest):bvy=sgn(ytest)*abs(bvy)+200}
 340   }
 350   bycnt=bycnt-1
 360   if bycnt=0 then {
 370     bvy=bvy+g:if bvy=0 then bvy=bvy+1
 380     bycnt=500/abs(bvy):if bycnt>11 then bycnt=11
 390     if bycnt=0 then bycnt=1
 400     by=by+sgn(bvy)*2
 410     if (mode=1)*(by<205)*(sgn(bvy)<0) then bvy=-bvy
 420     if (by>220)*(sgn(bvy)>0) then{ bvy=-bvy
 430       m_pcmplay(1,3,4)}
 440   }
 450   if (by>(31+high))*(by<(38+high))*(bx>21)*(bx<31) then {
 460     if sgn(bvy)<0 then goal=2 else if goal=0 then bvx=0:bx=
26:goal=1
 470   }
 480   if mxcnt=1 then mx=mx-1:if mx<15 then mxcnt=0
 490   mycnt=mycnt-1
 500   if mycnt=0 then {
 510    mvy=mvy+g:if mvy=0 then mvy=mvy+1
 520    mycnt=500/abs(mvy)
 530     if mycnt>13 then mycnt=13
 540     if mycnt=0 then mycnt=1
 550     my=my+sgn(mvy)*2
 560     if (my>200)*(sgn(mvy)>0) then mycnt=0
 570   }
 580   if (mx<15)*(mycnt<1)*((bx>60)+(by>130)) then sp_off(4):br
eak
 590   getkey=inkey$(0)
 600   if (getkey="0")*(mode=0) then mode=1:bxcnt=1:bycnt=2:mxcn
t=1:mycnt=0
 610   if (getkey="1")*(mode=1) then mode=2:bxcnt=0:bycnt=0:mvy=
-2000:mycnt=2
 620   if (getkey="2")*(mode=2)*(angle<9) then angle=angle+1
 630   if (getkey="3")*(mode=2) then { sp_off(4)
 640     m_pcmplay(0,2,4):mode=3:bxcnt=1:by=by-10
 650     bx=bx-5:bycnt=2:bvx=-1:bvy=-vector(2,angle) }
 660   if mode=2 then { bx=mx-10:by=my+3
 670     sp_move(4,bx-vector(0,angle),by-vector(1,angle),5) }
 680   sp_move(1,bx,by,1)
 690   sp_move(2,mx,my,2):sp_move(3,mx,my+16,4)
 700   /* for i=0 to 10000:next      /* 速度調整
 710 endwhile
 720 locate 10,5:if goal=1 then {
 730   m_pcmplay(2,3,4):color 1:print"GOAL!!":wincnt=wincnt+1
 740   sp_move(2,mx,my,3):sp_move(3,mx,my+16,4):color 3
 750 } else goal=0:print "NO GOAL"
 760 locate 5,10:print "hit space to end"
 770 while 1:getkey=inkey$(0):if getkey<>"" then break
 780 endwhile:if getkey=" " then break
 790 playcnt=playcnt+1:winrate=wincnt*100/playcnt:cls
 800 locate 8,0:print "成功率";wincnt;"/";playcnt;"(";winrate;"%)"
 810 endwhile
 820 end
 830 /************** subroutine ******************************
 840 func drawbg(high;int)
 850   line(3,20+high,23,40+high,41356):fill(3,1,25,110,0)
 860   line(23,20+high,3,40+high,41356):fill(24,high,25,50+high,
65535)
 870   box(3,20+high,23,40+high,41356):sp_move(0,26,40+high,0)
 880 endfunc
 890 func sprite_pattern()
 900   str temp[32]:dim char c(255):dim str chdat(8)[32]
 910   chdat={ "55662666366646605600200030004006",
 920           "52152515351545540202030303040404",
 930           "02020303030404040020103010401040",
 940           "00201030104010400002020303040400",
 950           "00020203031404000000201030114000",
 960           "00020203030404000002020303040400",
 970           "00002020004040000000002030304000",
 980           "00002020304140000000002030304000" }
 990   for i=0 to 7:temp=chdat(i):for j=0 to 31
1000     c(i*32+j)=temp[j]-48:next:next:sp_def(0,c)
1010   chdat={ "00000000000000000000000000000000",
1020           "00000000000000000000005150000000",
1030           "00000661660000000000646166500000",
1040           "00051661661500000006616161550000",
1050           "00016161615100000005111111150000",
1060           "00006161515000000000055155000000",
1070           "00000051500000000000000000000000",
1080           "00000000000000000000000000000000" }
1090   for i=0 to 7:temp=chdat(i):for j=0 to 31
1100     c(i*32+j)=temp[j]-48:next:next:sp_def(1,c)
1110   chdat={ "00011111111111000000111111111110",
1120           "00000811111111110000088881111111",
1130           "00000814881711110000081488177111",
1140           "00008888887771110000088888777110",
1150           "00000088877771100000000777777000",
1160           "88000488844449007880488884<488390",
1170           "78784444<48887397788<<<<4888779",
1180           "0777<<<<949888790077<<<<94988877" }
1190   for i=0 to 7:temp=chdat(i):for j=0 to 31
1200     c(i*32+j)=temp[j]-48:next:next:sp_def(2,c)
1210   chdat={ "00001111111111008080111111111100",
1220           "80801111111111008870111188111100",
1230           "87781188888811007708184088148100",
1240           "87808840881488008800888887888800",
1250           "88000888888880008880088644688000",
1260           "88800088668800007888<47777773770",
1270           "7888<<444433788778888<<<<<<48887",
1280           "0788<<<<<<<488870077<<<<<<<48873" }
1290   for i=0 to 7:temp=chdat(i):for j=0 to 31
1300     c(i*32+j)=temp[j]-48:next:next:sp_def(3,c)
1310   chdat={ "0000<<<<999988870000<<<999999887",
1320           "00009999999999887000099999999777",
1330           "00009999999999877000004444333770",
1340           "0000<<<<999999990000<<<999999999",
1350           "0000<<99999999999000004433944433",
1360           "00000887708887700000088770888770",
1370           "00000887708887700004636214464640",
1380           "00666661666664200443221333333220" }
1390   for i=0 to 7:temp=chdat(i):for j=0 to 31
1400     c(i*32+j)=temp[j]-48:next:next:sp_def(4,c)
1410   for i=0 to 255:c(i)=0:next
1420   c(1)=3:c(16)=3:c(17)=6:c(18)=3:c(33)=3:sp_def(5,c)
1430 endfunc
1440 func sprite_pallet()
1450   dim int data(15)={0,21141,35939,52851,65535,1345,1985
1460     ,38849,49089,63,63489,65473,41023,2047,31743,64809 }
1470   for i=0 to 15:sp_color(i,data(i),1):next
1480 endfunc
```


▶愛用していたX68000PROII(ルフィーア)がドライブの調子が悪くなったので，コンパクちゃんを連れ込んでしまった。ついでにMOも衝動買い。ごめんよルフィーア，ショートさせた私が悪いのさぁ～。
藤沢　実(20)東京都

ぷろぐらむ風まかせ（7）

➔コンパイルすると動かない！

さてさて，簡単に書けてインタプリタで楽々実行，コンパイルして.Xファイルにしてしまえばアセンブラや C で作ったプログラム並みのスピードで実行できるとても便利なX-BASICなんです。でも，書き方に気をつけないとインタプリタでは動いてもコンパイルすると動かないプログラムができてしまうんですよね。

なぜ，こんなことになってしまうかというと，それはX-BASICコンパイラのちょっと特殊な構造のせいなのです。

➔BASICコンパイラって？

普通，ほかの機種のBASICコンパイラなどではBASICのコンパイラだけで単独のパッケージになっているんですが，X-BASICのコンパイラは「C Compiler PRO-68K」のパッケージに入っています。で，プログラムをコンパイルするには，

A>CC プログラム名.BAS

と C のプログラムをコンパイルするのと同じようにするんですよね（C の場合はA>CC プログラム名.C）。

実はX-BASICコンパイラは C コンパイラと同居して，C コンパイラの力を借りてプログラムをコンパイルしているのです。たとえば，先ほどのプログラムであれば，まずCC.Xという C コンパイラドライバがプログラム名.BASがBASICのプログラムであるかどうかを判定してくれます（拡張子を見ているだけですが）。で，BASICのプログラムであれば，C に書き直します。それから，C のプログラムとしてコンパイルしてくれるのです。ちなみに C コンパイラも C のプログラムをアセンブラのソースに直して，.Xのプログラムにしてもらっています。つまり，X68000では，図 1 のようになっているわけです。

➔マニュアルに書かれていること

BASICと C のコンパイラが一緒になっているとメリットもあるのですが，いろいろと制約も出てきます。「C Compiler PRO-68K」の C ユーザーズマニュアルを見てください。ver.2.0以降のマニュアルであれば115ページあたりから見てください。

主なところでは，コンパイルするプログラムでは名前（変数や関数）のつけ方に制限があったり，複数の演算がひとつの式に出てきた場合に演算順序をはっきりさせる必要があったりすることです。ほかにも，X-BASICと C の論理演算の値の違いについても注意が必要ですよ。

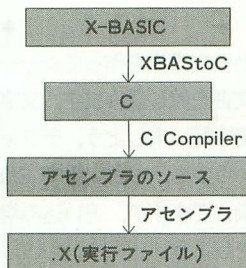



図 1

➔BCチェック

さてさてさて。

マニュアルにも 4 ページに渡って書いてある注意事項を手作業でプログラムリストから見つけ出して修正を加えるのも，ちょっと大変な作業ですよね。実はこの作業に，とっても使えるプログラムがあるのです。シャープから発売されている「XBAStoC CHECKER PRO-68K」がそれです（まだ手に入るだろうか？）。たとえば，今月号の「2DPLANT.BAS」の元のプログラムでこれを使うと，リスト1，2のようになりました。

1030行を見てください。DDR1という関数が呼び出されています。この関数は1330行に定義されています。

これを見るとわかるように，4 番目の引数であるhaはデフォルトのint型であるのに，呼び出した側は「1+fj*((bd(0,o1,o2)/5)+1)*1.3#」とfloat型です。C のプログラムでは，小さい型の変数に大きな型の定数や変数の内容を代入したときには，キャストといって，「小さいほうの大きさに合わせる」と宣言しないとその内容は保証されません。ですから，DDR1を呼び出すときに 4 番目の引数は，

int(1+fj*((bd(0,o1,o2)/5)+1)*1.3#)

としてint型にしてから渡せばいいわけです。オリジナルではここで引数の型変換に失敗して，DDR1関数の内部でループカウンタがマイナス何百万やプラス何百万という値になって，ほとんど無限ループに陥っていたと思われます。

そうそう，これは経験上から思うことなのですが，変数や引数にあまり長い式を代入すると，思ったようにキャストしてくれない，ということが多いように思います。式を使うときにはできるだけ短く小刻みに書き，わかりやすい式を心掛けたほうがいいと思います。

リスト 1

```
[ERROR LINE ]

 1000        if hhp1<>0 then {
 1010          for fk=1 to hhp1
 1020            sa1=asc(mid$(hhp(hpd1,rr(fj),fi),((fk-1)*3)+2,1)
)-48
 1030            DDR1(o1,o2,sa1,1+fj*((bd(0,o1,o2)/5)+1)*1.3#,hv(
fj),1.7#,2,3,o4)
                                                           ▲( 20 :
注意 )
 1040        next }
 1050      }
 1060    next
 1070    if fi=0 then {
 1080      if o4=0 then kkx=236:kky=210
                       ▲( 33 :注意 )
                              ▲( 33 :注意 )
                                   :
                                   :
                                   :
                                   :
 1330 func DDR1(o1,o2,sa;float,ha,va;float,ha2;float,oc1,oc2,o4)
:/** 画面描画 **
                 ▲( 17 :エラー )
                    ▲( 17 :エラー )
                              ▲( 17 :エラー )
                                                      ▲( 17 :エラ
ー )
                                                         ▲( 17
:エラー )
                                                            ▲(
 17 :エラー )
```

リスト 2

```
[ERROR REPORT ]

000103  (01030行)  20:注意   int型関数の引数にfloat型の値を渡した
000108  (01080行)  33:注意   未定義の外部関数および変数名
000108  (01080行)  33:注意   未定義の外部関数および変数名
                                     :
                                     :
                                     :
000133  (01330行)  17:エラー 仮引数の型が宣言されていない
000133  (01330行)  17:エラー 仮引数の型が宣言されていない
000133  (01330行)  17:エラー 仮引数の型が宣言されていない
000133  (01330行)  17:エラー 仮引数の型が宣言されていない
000133  (01330行)  17:エラー 仮引数の型が宣言されていない
000133  (01330行)  17:エラー 仮引数の型が宣言されていない

   "2dplanta.bas" には、164個の問題点があります。
```

▶予言「X68000の未来はCD-ROMの普及にかかっている！」かもしれない（苦笑）。
伊藤　勝利（31）千葉県

新製品紹介

XDTP SX-68K

NakanoShuichi 中野 修一

SX-WINDOWで初めてのDTPソフトの登場です。アウトラインフォントを同梱していますので出力クオリティも十分確保できます。新しいIVMの機能により，JGフォントなら半角文字もアウトラインフォントで美しい出力ができます。

DTPといっても普通の人にはあまり縁のない分野かもしれません。

個人で本を作るということは滅多にないことですし（同人誌を作るためにXDTPを買った奴もいるが……），普段の文書ならシャーペンでもまあまあ間にあいます。

要は，文字（特にアウトラインフォント）と図形を自在に組み合わせるツールだと思えば，大がかりなものではなくても，ちょっとしたパンフレットや，ちょっと凝ったレポート作成などに使うことも考えられます。身近なところでは，年賀状を作るときにもある程度使えるかもしれません。

これまでにも，こういった指向を持ったソフトがいくつか出てきていましたが，それらと比べると，かなり「DTPソフトらしく」なっているといえるでしょう。

そもそも，他機種ではパソコンの用途として，「出力/レイアウトを考慮した文書作成」というものが一定の位置を占めているのに対し，X68000ではこういった分野はほとんど未開拓の状況です。

完成された文書を作るために「見出しなどレイアウトをしたい」とか「段組み印刷をしたい」とか「図版を張り込みたい」といった要求があっても，なぜか，低い操作性と豊富な網掛けを備えたワープロのようなものしかサポートされてきませんでした。

今回紹介するXDTPはSX-WINDOWシステムを使ったDTPソフトです。本格的な用途に使うには力不足なところも見受けられますが，シャーペンではうまくできなかったことが，わりと簡単に処理できるようになります。

このページは版下としてXDTPの出力をそのまま使っています。360dpiのインクジェットプリンタで出力したものを原寸で使用してみました。版下には品位が低いので，お見苦しい点もあると思いますがご了承ください。

使ってみよう

まずシャーペンなどで文書を作成します。次に紙の上に適当でかまいませんからラフレイアウトを作成します。ページ単位でどこをどのようにしたいのかを頭に入れたらXDTPを立ち上げます。

XTDPのユーザーインタフェイスは設定によりツールバーとフローティングメニューの2種類が選択できます。

フローティングメニューは開いた瞬間に素早くマウスを動かすことでウィンドウとして分離することができます。また，分離したメニューはクリックで下層のメニューを開き，プレスでポップアップさせることができます。カチっと押すと分離したメニューがポツリと開き，しばらく押すと通常のメニューのようなものがだらっと現れるのです。

実はツールバー形式にしておけばこれらは両方同時に使用できます（分離したメニューから親メニューを呼ぶ）。なら，設定は絶対ツールバーにしておくべきだ……と思った人はいませんか？

そもそも，どうしてメニューバーを消したりできるかというと，それはおそらく100%表示にしたとき，A4用紙の縦いっぱいが1画面に収まるようにするためだと思われます。ルーラーかメニューを消せばなんとか入ります。というわけでここはメニューバーを消すしかないでしょう。

XDTPのユーザーインタフェイスのデキはフローティングメニューの双肩にかかっているわけですが，反応がちと遅いのが気になります。ボタンを押してクリックでないのを確認してポップアップが始まるのです。こういうものはシステムで設定されているダブルクリックの判定時間をもとに時間設定されるのが常識ですが，これはそのようにはなっていないようです。

*　*　*

とにかく文書枠を作ってシャーペンでセーブした文書を流し込みます。文字の大きさや各種設定を行いましょう。デフォルトで禁則なしというのも，ちと大胆ですね。禁則はいろいろ揃ってますが，追い込み均等が一般的でよいでしょう。

ただ，禁則された文字の後ろに改行があると改行だけ次の行に送られてしまうという近

マスターフォームとはなにか？

XDTPの文書は2層に分かれています。すなわち，マスターフォーム層とドキュメント層です。これらは互いに独立したもので，ページ内で見ればまったく同じ機能を持っています。どう違うかというと，マスターフォームは右ページ/左ページの基本設定があるだけなのに対し，ドキュメントは各ページで自由な内容を設定できることです。逆にいえば，ドキュメントはページごとに設定し直さなければなりません（レイアウト含む），マスターフォームは設定されたものをそのまま流用することができます。前のページからコピーしてくる手間が省けますから便利なこともあるのでしょう。

常識的なDTPソフトではマスターフォームに設定された基本レイアウトが新しいページを作るときにドキュメント部に流用されるのですが（文書を流し込めば文書量に応じてページが作成される）XDTPではページ単位でレイアウトを作り直し，前のページとのリンク指定を手動で設定しなければなりません。

いちいち文書ボックスを前のページと同じになるように作成するのはかなり骨の折れる作業です。対策としては，基本レイアウトをマスターフォーム側に作成しておき，新規ページを作るたびにドキュメント層へコピーしてやればいいでしょう。

この場合，マスターフォームを使う意味がまったくなくなりますので，「マスターフォームを使用しない」設定にすれば多少は処理速度を上げることもできます。

年珍しくなった仕様を持っているため，なにも考えずに設定をすればいいというわけにはいきません。手作業の調整は必須でしょう。

私が確認したところでは，もっとも綺麗に文字を並べる方法はシャーペンで作成した文書を改行挿入にしてファイルに書き出し，連続部分は均等割り付け，余った部分は左詰めで処理することです。

手を加えればかなりよくなるのですが，やはり改行記号を1文字とみなす仕様のために調整が難しい部分も多々あります。均等割り付けなどは，なにも考えないと，

サ　ン　プ　ル
テ　ス　ト　テ　ス　ト
均 等 割 り 付 け の テ ス ト

のようになります。本来なら，

サ　　ン　　プ　　ル
テ　ス　ト　テ　ス　ト
均 等 割 り 付 け の テ ス ト

のようになってほしいところなのですが。

以上，SX明朝でお送りしました。

ここからはツァイトのJGフォントから細明朝体Bでお送りします。IFM ver.4になったことで，ツァイトのJGフォントを使用している場合に限り，半角アウトラインフォントが使用できるようになりました。

JGフォントにはWindows用に3書体入って普通のフォントの半額というお得なパックが発売されていますが，できれば単体売りのものを購入されることをおすすめしておきましょう。フォントの品位がまったく違います。ともかく，XDTPがちゃんとアウトラインフォントをつけてきたというのは評価できます（第一水準だけだが）。

XDTPでは縦書きもできます。

SX-WINDOWでXDTP SX-68Kを使用して縦書き印字を行ってみました。半角文字の並びが嫌な感じですが，カンマくらいなんとか…。しかし_main.0とか，(10+15)/0.25=100なんてやるのは酷というものでしょうか。

といった具合です。

混迷する単位系

XDTPではときどき文字枠の大きさの調整がうまくいかないことがあります。文字に限ってはすべてのものが1ポイント単位で制御されるというのに問題があります。

その昔，MacintoshのDTPが使いにくいながらも実用にされていたのは，文字などの大きさがポイント指定しかなくても，ちゃんと小数点以下まで見ていたことが要因のひとつと思われます。

XDTPでは文字の縦横比をパーセント単位で指定できるのですが，たとえば9ポイント文字の場合，

100%　東京特許許可局
99%　東京特許許可局
89%　東京特許許可局
88%　東京特許許可局

となります。

ちょっと謎なのは，文字に関する設定，たとえばサイズや字間などの指定もすべてポイント指定なのですが，なぜか縦横比だけはパーセント指定になっています。しかし，すべての単位が1ポイントでクォンタイズされていますから結局は1ポイント単位の精度と同じことになります。

じゃあ基本文字サイズを変えたときに有効になるのかというと，そういった設定を変えるとほかのアトリビュートはすべてデフォルトのものに再設定されてしまいますので，ますますわけがわからなくなっています。

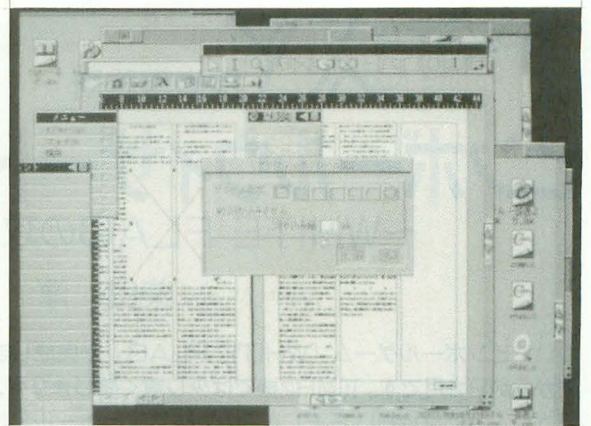
たとえば400%に拡大した場合，画面上の1ドットは1ポイント以下の大きさになるのですがこの状態でもちゃんとマウスはドット単位で指定可能です。フレーム枠などの指定は1/4ポイント単位での精度があることになります。

さらに文字の回り込み指定などではミリメートル単位の指定を行います。ここではポイント単位の指定はできません。

現実問題として，ポイント（0.3514㎜単位）という単位は日常生活とはかけ離れたものだと思います。

もうひと工夫必要か？

XDTPでなにより致命的なのはドローデータが張り込めないことでしょう。また，シャーペンで書いた文書をコピーしてペーストすることすらできないのはちょっと問題があるのではないでしょうか。

「CANVASPRO-68KやPrintShopPRO-68Kのデータが使用できます」といっても，わざわざSX-WINDOW環境を降りて図形をエディットする

各フレームで回り込みの指定ができる

くらいなら，マルチウィンドウ環境なんて必要はありません。

また，あいかわらずSX-WINDOWの色変換は驚くほど遅く，変換結果は汚いものです。

XDTP上では画像枠だけとっておいて，そこに別のツールで印刷したものを切り貼りするのが正解でしょう。それなら処理もさほど重くならず，文書ファイルも抑えられます。

操作性の面でもいくつか腑に落ちない動作がみられます。

たとえば文書のスクロール。画面書き換えのときにカーソル位置を中心にして書き直します。気を利かせた処理のようにも思えますが，要するにそこで行われる作業がなにもわかっていないということです。文書をスクロールするときはカーソル付近の情報が見たいわけではありません。カーソルの前後の情報が見たいのです。さらにカーソルのカラム位置は保存されませんから，前後を参照したいと思ったときはわざわざスクロールバーに頼らなければならなくなります。

技術的に困難なことは要求しませんが，画面で文字色が確認できないとか，見開きの左側を奇数ページと呼ぶとか，吸着なしでないと微妙な操作ができないとか，デフォルトが24ポイント（8.4336㎜）の大きさの文字だとか，印刷倍率がないとか，結構常識的な部分で問題があるように思えます。うーむ，そこそこのことはできるんですがねぇ。

最初に書いたように，XDTPはSX-WINDOWシステムを使ったDTPソフトです。今度はちゃんと「SX-WINDOW上のDTPソフト」と表記できるツールを希望したいところです。

XDTP SX-68K　35,000円（税別）
シャープ　03(3260)1161

こちらシステム X 探偵事務所

FILE-XVII

illustration : T. Takahashi

ピンボールウィザードへの道
～WHITE FLAGの攻略～

Shibata Atsushi 柴田 淳

ついにピンボールゲーム「WHITE FLAG」が完成しました。制作に取り掛かってから約9カ月です。10MHz機では少し重く感じますが，まずは一度遊んでみてください。今回はWHITE FLAGの攻略方法とルールの説明などです。

自分で作ったゲームの攻略を自分で書くなんてナニだが，まあこういうかたちもある意味では理想的ではある。なにしろ，ただ遊んでいるだけではわからない裏の裏まで知っている人間が，ゲームについて書こうというのだ。

ところで，WHITE FLAGとは，モトクロスレースの最終ラップに振られる旗のこと。絵柄とか役ものの名前などもそれっぽくしてある。実際のピンボールもそうだが，デザインや音だけでなく，ルールなどもゲーム全体からひとつの雰囲気をかもし出すのがピンボールの王道なのである。

と，前置きはこれくらいにして，遊ぶときのコツを，プレイヤーの熟練度ごとにまとめてみたい。

初級・ミニゲームを狙え

このゲームでまず身につけてほしいのは，フリッパーを上げ，フリッパーの根元にボールを止めた状態（ホールディングと呼ばれる基本技）に確実にもっていけるようになること。そこから，確実にボールをループできるとなおよい。

マルチボールゲームのジャックポットボーナスも確かに魅力的だが，無理にアクセルターゲットを狙っているとボールはたやすく落ちてしまう。初めのうちは無理をせず，とりあえずループを確実に取る訓練をしよう。

3回ループに成功すると，ミニゲームの権利が発生する。ミニゲームとは，制限時間内に条件を満たすなどすると，ボーナスが得点に加算されたり，ゲームが進めやすくなったりする特別ルールだ。

ルールを見るとわかるとおり，ミニゲームは権利発生後，数種類の条件を満たすことで始まる。それらミニゲーム開始のキーとなる仕掛けのなかでもいちばんおいしいのが，やはりクイックターゲットだろう。たとえば，制限時間のないミニゲームを，クイックターゲットを作動させて取ったとすると，ミニゲームの権利はそのまま保留される。クイックターゲットに入ったボールは上部階層に打ち上げられ，そのまま右ランプを通るので「リスタートミニゲーム」というルールが適用されるのだ。また，取ったミニゲームがクイックマルチボールなら，そのまま2ボールのマルチボールプレイとなる。

ついに完成したピンボールゲーム「WHITE FLAG」

要するに，狙ったところにボールを打ち上げられるようになることが出発点なのだ。マルチボールターゲットばかり狙うより，ボールを確実にループさせ，1つひとつミニゲームを取っていくほうが，ゲームの組み立て方としてはスマートといえる。

中級・デスコースを見抜け

確実なホールディングをものにし，ループもバシバシ決められるようになると，ボール1（1個目のボール）をプレイ中にミニゲームのライトが点滅しているだろう。このくらい操作に熟練すると，今度はいかに1個のボールで長くプレイするかにゲームの成否がかかってくる。左のアウトレーンに落ちたボールはこまめに左ランプにボールを通していればキックバックが押し戻してくれる。問題は右のアウトレーンとフリッパーの間を落ちていくボールである。

ピンボールでは，ボールが落ちる典型的なコースを「デスコース」と読んでいる。どのような場合にボールが落ちやすいかを見抜いて，そのようなコースはなるべく避けるか，防御手段を行使する。もちろん，デスコースは台ごとに違うので，いつも同じ手が通じるとは限らない。

WHITE FLAGにもデスコースがいくつかある。プレイフィールド右中央の，右ターゲットの上の出っ張りに緩い速度で当たったボールはわりとフリッパーの間に落ちていきやすい。また，右スロープを上がり切らずに戻っていくボール，左ループの入口に入って戻ってくるボールも同様。そのほか，スリングショットの外側に当たるボ

ールはアウトレーンに落ちやすい。

このように，落ちやすいコースをたどるボールには「揺らし」で対応する。それもボールがデスコースに入る前に台を揺らす。揺らしには「横揺らし」と「縦揺らし」の2種類があるので，適宜使い分けるようにする。

アウトレーンに向かうボールは，縦揺らしと横揺らしを併用することでたいていリターンレーンに送り込むことができる。問題はフリッパーの間を落ちるボールをいかにうまく処理するかである。

上級・究極のボーナス

WHITE FLAGのルールは，マルチボールプレイとミニゲームの同時進行を許す。特にデッドヒートやウォッシュボードなどのミニゲームが進行中にマルチボールが始まると，その相乗効果は絶大だ。そのためには時間を多く取りたい。制限時間のあるミニゲームが始まったら，右フリッパーからモアタイムターゲットに数回ボールを当てておくのが理想である。

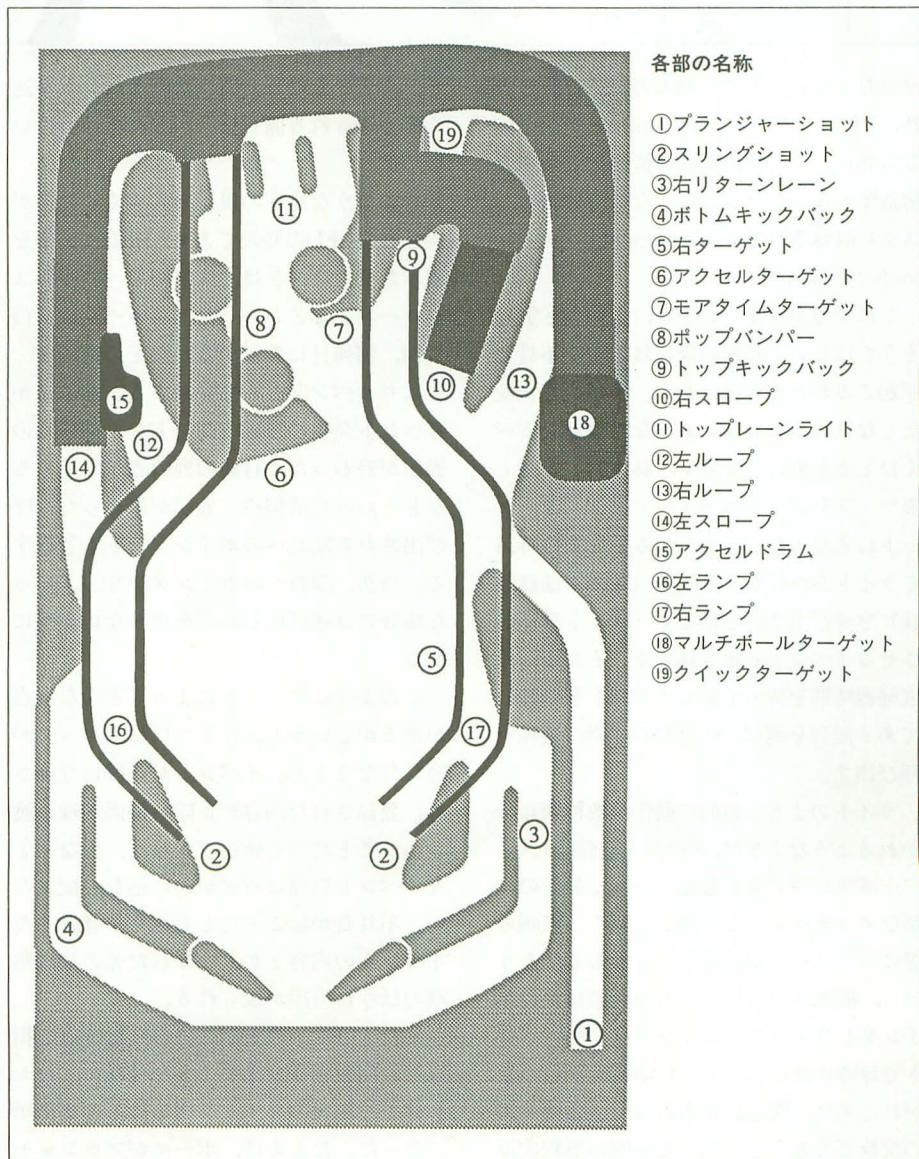
と，ゲームに慣れてくると，自分なりにゲームの組み立てを考えながらプレイできるようになる。しかし，どのような組み立てであろうとも，目指すところはただひとつ。ミニゲームをすべて取り，その後のマルチボールでジャックポットを完成させること。コレクトエブリゲームからジャックポットを完成させると，プレイヤーはスコアマルチプライヤーを得ることができる。なにやらものものしい名前のボーナスだが，要するにこれはゲーム終了後，プレイヤーのスコアを倍加してくれるものだ。

このボーナスを取れば，数億点のスコアも夢ではない。なおスコア用の変数には符号なしのlongがとってあるので，桁溢れを心配することはない。

開発を終えて

実は，大部分の開発はMacintoshで行った。コンパイル速度だとか，絵柄の編集効率などを考えてのことだった。

ゲームに使ってあるピンボール台の絵は，いきなりこのように描き込んだものではなく，バンパーやスロープといったパーツごとにペイントデータを編集して，最後にひとつに組み上げたものである。こうでもしなければ，ろくにCGも描いたことのない僕がこんな絵柄を描けるはずがない。こういう類のものは，絵心以上にこまごましたテクニックを要求されるものなのだ。これらのテクニックを，大きなメモリとハイパフォーマンスなマシンパワーで吸収してくれるのがMacintosh，というわけである。

で，Macintosh上で作ったプログラムはX680x0に移植されるわけだが，その際に問題となるのは主にグラフィック描画とサウンドだろう。MacintoshのプログラムはSX-WINDOWのプログラムと同じように，イベントループを核として処理を枝分かれさせるような構造をしているのだけれど，ゲームの本体部分はほとんどイベントループからは解き放たれているので，イベント処理は単純に切り放すだけで済む。

問題の2つのうち，サウンドはMacintoshではまったく鳴らさなかったのでX680x0のほうで新しく作ることになる。問題は，描画処理をどう移植しようか，ということだ。まず，Macintoshで描画によく使われるCopyBitsという矩形領域の画像を転送するルーチンがある。これの下位互換のルーチンを自前でこしらえる。

また，CopyBitsには描画環境の先頭番地，1ラインのバイト数などを管理する構造体へのポインタが渡される。MacintoshのグラフィックエンジンであるQuickDrawでは，全般的にこの情報を基に描画処理を行っている。つまり，示された先がG-RAMであろうがメモリ領域であろうが，

台の説明

ソツなく受け入れてくれるのだ。だから当然，下位コンパチのCopyBitsルーチンを作る以外に，この描画領域の情報を管理する構造体も作らなければならない（付録ディスクに収録されているmacindef.hとmacintools.cを参照）。

これで万事うまくいくか，というと実はそうではない。移植の際にはどんな不具合が起こるかわからないから，移植の際に変更しなければならないような処理はなるべくひとまとめにしておく。具体的にいうと，裏マップを調べ，ボールがアクセルターゲットに当たったことがわかると，その時点でライトを点灯させる処理（つまり描画処理）を呼び出すのではなく「ライトを点灯させるイベント」を登録する。そして，垂直帰線時間を待ってからイベントを登録してある配列を調べ，必要なら描画の処理を呼び出す。

ライトのように目的の動作が数種類に分かれるようなときは，イベント登録時に「3回点滅後に消える」などといった動作の種類をメッセージとして渡しておく。描画の際にパラメータが必要なこともあるだろうから，描画処理にはメッセージのほかにポインタも渡す（このポインタは当然イベント登録時に渡されたもの)。ポインタにつながれた先は，構造体であろうとlongサイズの変数であろうと，描画処理側の解釈次第でどうとでもなる。要は描画イベントを送る側と送られる側で整合がとれていればいいのだ。

このような処理の規模を大きくしたのが画面右のLEDの処理である。LEDに文字を表示させたいときは，文字列と「右からスクロール」などといったアニメーションの種類，何列目に表示するかなどの情報を，やはりイベントとして登録する。そのほか，イベント発生の遅延時間だとか，何列目の表示が終わったら目的の列の表示を始めるなどといった情報や，表示が終わったら呼び出される関数へのポインタなども登録する。当然，関数へのポインタがNULLだった場合には呼び出し処理を行わないようにする。

このようにすることによってどんな利点があるかというと，ひとつにはデバッグが容易になること。イベントが正確に登録され，登録された内容が正確に描画処理に渡されることだけを確認できれば，少なくともイベント処理にバグが紛れ込む心配はない。不具合が起こったとしたら，登録したイベントの内容とか，呼ばれた先の描画処理のほうに原因が絞られる。

また，LEDへの表示が終わった時点で指定した関数を呼び出すような仕様にしたおかげで，処理のタイミングを計るのが簡単になった。たとえば，ボールがプランジャーショットに打ち出される処理などは，LEDのメッセージの何行目かが表示されたあとに呼び出されている。表示が始まる前とか終わったあとというのではなく，表示が行われている最中にボールが現れるのだ。こっちのほうがなんとなく「カッコイイ」ではないか。

開発が終わって，ソースのサイズを計算してみてびっくりした。とはいっても総量で200Kバイトくらいで，SLASHの足元にも及ばないのだが，これだけのサイズのソースをひと文字ひと文字ポチポチと，それも自分ひとりで打ち込んだのだと客観的に振り返ってみると，やっぱり驚く。

で，これだけのサイズのゲームを，たとえばLEDなどの処理をするにしても，グローバル変数を乱発してゴテゴテのソースを書いていたら，きっといまだに完成しないで，編集さんに泣きついていたのではないだろうか。処理の流れをイベントとして管理するような手法を使ってさえ，かなりギリギリの状況までコードを吐き出していたのだ。想像するだけで空恐ろしい。

いままではとにかく動くソースを書くのが精一杯で，開発の効率とか保守性などを考えていられなかった。C言語を始めて2年弱，ここへきてやっとまともなソースが書けるようになってきた。数値をコードに埋め込むこともあまりなくなったし（でもまだたまにやる)。

ところで最近，ピンボールを置いてくれるゲームセンターが家のまわりにいくつかできて，大変うれしい。で，そのうちのひとつに，なんと往年の名作ピンボールビデオゲーム「ピンボールアクション」が置いてあった。そこは，ほかにもいくつか懐ゲーの類が置いてある店なのだが，それにしても，目の付けどころがシブくてよい。ちかごろはゲームセンターに行っても，ピンボールばかりやってビデオゲームには手を出さないのが常となっていたのだけれど，そのときばかりはさすがに違った（とはいっても結局ピンボールではある)。

久しぶりにプレイした「ピンボールアクション」は，記憶していた以上によくできたゲームであった。自分で作ってみたジャンルだけあって，ゲームの細部まで作り込んであるのがとてもよく感じられる。強いていえば，スリングショットにボールが当

たったとき，当たる場所によって押し出される角度が変わらないのが気になったが，それ以外欠点らしい部分は見あたらない。「ピンボールアクション」といえば，もう10年近く前，テーカン（現テクモ）が作ったビデオゲームだ。そう，10年も前のゲームなのである。

その日の僕は，自己嫌悪と興奮がごっちゃになったような不思議な感覚を抱きながら家路についた。プログラミングにしてもゲーム作りにしても，「日々是精進」なのである。　つづく

WHITE FLAGの遊び方

Sキーを押すと，プランジャーレーンにボールが現れる。これをプランジャーショットで打ち出し，プレイフィールドに送り込むことによって，ゲームが始まる（キー操作は表1を参照）。

ゲーム中はフリッパーでボールを弾き，さまざまな仕掛けを作動させることによって得点を得る。また，ゲームの状態は大まかに「シングルボールプレイ」と「マルチボールプレイ」に分かれ，おのおののプレイ状態はさらに「ミニゲーム」と「ノーマルゲーム」に分かれる。プレイ中どの状態にあるかによって仕掛けの作動のさせ方や得点などが変わってくる。

すべての状態に共通のルール

・RETURN TO THE HEAT

以下の条件でボールがアウトレーンに落ちたときは，自動的にプランジャーショットが作動する。

（1）プランジャーショット後，ボールがフリッパーに触れずにアウトレーンに落ちた場合。

（2）プランジャーショット後，フリッパーにボールが触れてから一定時間内にボールがアウトレーンに落ちた場合。

（3）トップキックバックにボールが押し戻されてから一定時間内にボールがアウトレーンに落ちた場合。

・RESTART

マルチボールでないときに左リターンレーンのランプが点灯していれば，ボールはボトムキックバックによりフィールドに押し戻される。

・LAST CHANCE

左リターンレーンのランプが消えているときに左ランプレーンにボールを通すと，ランプが点灯する。

・LOOP BONUS

左右のループを通すごとに5万点のボーナス。

・QUICK LOOP

ループ後すぐにボールをループさせると，100万点のボーナス。

・HAIRPIN BEND

トップキックバックからボールが跳ね返った直後にランプレーンを通すと，100万点のボーナス。

・COLLECT TOPLANE RIGHT

トップレーンの3つのライトをすべて点灯させるとドロップ後のボーナスが倍加される。

マルチボールについて

・ACCELE

アクセルターゲットのランプを3つとも点灯させると，左スロープに仕掛けられたアクセルドラムが一定時間作動。アクセルドラム作動中にボールを左スロープに入れるとボールは加速し，マルチボールターゲットに導かれる。

・MULTIBALL

マルチボールターゲットに3回ボールを入れると，3ボールのマルチボールとなる。

マルチボールのルール

マルチボールプレイの最中は，クイックループのボーナスは加算されない。その代わり，以下のマルチボールルールが適用される。

・ADD A BALL

クイックマルチボールでないマルチボール中，左ランプレーンにボールを通すたびにフィールド上のボールが最大3つまで増える。ただし，3つ目を獲得した時点で権利がなくなる。

・INSTANT EXTRA BALL

マルチボール中，エキストラボールの権利を得ていない状態で左右どちらかのアウトレーンにボールを通すと，トップキックバックのエキストラボールライトが点灯しエキストラボールの権利を得ることができる。

・JACKPOT

マルチボール中，アクセルターゲット，右ターゲット，および3つのランプレーンすべてを点灯させることによりアクセルが回転。そのあとマルチボールターゲットにボールを入れると500万点のボーナス。

・SUPER JACKPOT

ジャックポット獲得後，同一条件を再び満たすと1000万点のボーナス。

ミニゲームについて

ミニゲームの権利がない状態から，以下の条件を満たすとミニゲームの権利が発生する。

（A）ボールを3回ループさせる。

（B）ボール3をプランジャーショットした後。

権利発生後，ミニゲームライトが点灯し，左右のスリングショットに当たるたびにライトは（a）～（g）へと移動していく。

そして権利発生中に，以下の条件でミニゲーム（a）～（g）がスタートする。

（1）クイックターゲットにボールを入れる。

（2）3つの右ターゲットのランプをすべて点灯させ，左右どちらかのランプレーンにボールを通す。

（3）3回ボールをループさせ，トップキックバックを作動させる。

制限時間のあるすべてのゲーム中に，（1）以外の方法でミニゲームを再び発生させると300万点のボーナス。ただし，（2）は左のランプレーンを通さないと作動しない。また，クイックターゲットにボールを入れると100万点のボーナス。

（a）DEAD HEAT

指定時間内にポップバンパーに10回ボールを当てると100万点のボーナス。以後ボールが当たるたびに5万点。また，ポップバンパーに10回ボールを当てたあとボールが10回バンパーに当たるたびに，バンパーの得点が倍加する。最高倍率は16倍。

（b）WASH BORD

時間内に3つのランプターゲットにボールを当てるたびに100万点のボーナス。

（c）ANOTHER LAP

トップキックバックにボールを当てるとエキストラボール。時間無制限。すでにエキストラボールを得ていたときは100万点のボーナス。

（d）OVER THE TOP

右ランプレーンにボールを通すと2ボールのクイックマルチボール。時間無制限。

（e）FULL THROTTLE

指定時間中，ループさせるたびに50万点のボーナス。

（f）HIGH JUMP

アクセルターゲットをそろえたあと，ボールがロックされるまでアクセルドラムが作動し続ける。マルチボール獲得まで有効。

（g）OILING

アクセルドラムの作動時間が2倍に延長される。ゲーム中有効。

また，ミニゲーム中には以下のイベントが複合して起こる。

・RESTART MINIGAME

ミニゲーム獲得後，制限時間中に以下の条件を満たすと再びミニゲームの権利が発生する。

（1）同一ボールでのループ。

（2）左右のランプレーンのどちらかにボールを通す。

・MORE TIME

制限時間のあるミニゲーム中，モアタイムターゲットにボールを当てると制限時間が延長される。

・COLLECT EVERY GAME

（a）～（g）まですべてのミニゲームを獲得すると，マルチボールゲームが開始される。このマルチボールゲーム中，ジャックポットを獲得すると，ゲーム終了後，得点が倍加される。倍数は16倍まで上昇する。

表1　キー操作

スタート	S
左フリッパー	XF1
右フリッパー	XF5
プランジャーショット	SHIFT
台の揺らし（縦方向）	SPACE
台の揺らし（横方向）	XF3
ゲーム終了（常時）	ESC
・ハイスコア登録時	
文字を1つ戻す	XF1
文字を1つ進める	XF5
文字を1文字登録	SHIFT
文字を1文字削除	DEL
登録名を決定	RETURN

TEX入門講座 ～てふてふらてふ～

表の作成

Taki Yasushi 瀧 康史

今月は表を作成していきます
それほど難しいことは行っていないのでいろいろ試してみましょう
これでレポートの作成もばっちりですね

やっぱり，愛。愛だよなぁ。

なにがって，この本が発売される頃には，あの「超兄貴」の続編「愛・超兄貴」が発売されるのさ。うーん，2年もファンを待たせるんじゃないぞ。兄貴なスタッフさんたち。

話は変わって秋はやっぱりポージングっすよね。ポージング。私の世界情勢の最大の関心は，1994年オリンピア。さすがに，海の向こうまで行って見てくるほどお金持ちじゃないけど。来月の『IRONMAN』は発売日に買わなくっちゃな。うん。

前回2回で，TeXに慣れ親しむためのアプローチと，理系レポート必須の数式処理を書くための方法について書いた。とりあえず目的のひとつである，「理系レポートや論文をTeXで処理すること」を達成するには，あとは表の作成が必要。でもって，たいていのレポートをTeXで処理することができるようになるためにも，今月は表の作成について，話を進めることにしよう。

tabular環境

当然ながら，システムはLaTeX。plainTeXでは非常にややこしい表作成も，LaTeXなら表に関するマクロが充実しているため，かなり簡単に利用できてしまう。

LaTeXの上で表を作成するには，2つの環境を利用する。まずひとつはtabbing環境，そしてもうひとつはtabular環境。tabbing環境は名のとおり，TABを利用して表を作成する方式なので，罫線を含むような複雑な表は書けない。だからといってtabbing環境が特別に楽というわけではないので，私は通常tabular環境を利用している。それで困ったこともないので，まずはtabular環境について説明することにしよう。

2×2の簡単な表を印字例2の1.1に記してみた。tabular環境を素直に使うとこうなる。まずはこのtabular環境についてのルールをマスターしてもらおう。

¥begin{tabular}～¥end{tabular}までネストされている。この環境は，

¥begin{tabular}[表位置の指定]{項目の指定}
　要素 & …… & 要素 ¥¥
　要素 & …… & 要素 ¥¥
¥end{tabular}

と，こういう構成。

それぞれについて説明しよう。

まずは表位置の指定。

印字例2の1.1では無指定だが，[t]や[b]で位置の変更ができる。まあ，この指定は使わなくても問題ないだろう。

次の{項目の指定}は，表の横の要素を決定する。{}の中には，主に，|，||，l，c，rを利用し，1.1の例でわかるとおり，|は縦線の指定，cはセンタリング指定だ。これがlになれば左寄せであるし，rであれば右寄せ。||にすれば縦棒が2本になる。1.1では，まず左に縦の線を引き，そして表の第1項目をセンタリング。線を引かずに第2項目をセンタリング。そして右側に線を引くという感じだ。二重の縦線は項目を分けるときに使うとよいだろう。

定義(21行目)を行ったら，あとは項目の要素を書く。項目の要素は&で分けられ，この数は，{項目の指定}の中のl，c，rの数と等しくなくてはいけない。表の行末は必ず¥¥で終わること。

印字例1

表

瀧 康史

1994'10/13

目次

1 tabular 環境の簡単な使い方 2
　1.1 単純に tabular 環境のみ使う 2
　1.2 表を飾る 2

2 tabular 環境ですこしテクを 2

3 tabbing 環境 3

表目次

　1 身の回りの CD-ROM ユーザーの数 2
　2 私の周辺から無作為に 10 人選んだときの 68 ユーザの状況 2

リスト1

```
 1: ¥documentstyle[12pt,a4j]{jarticle}
 2:
 3: ¥title{表}
 4: ¥author{瀧 康史}
 5: ¥date{1994'10/13}
 6:
 7:
 8: ¥begin{document}
 9: ¥maketitle
10: ¥tableofcontents
11: ¥listoftables
12: ¥vfill¥eject
13:
```

表の途中で横線を引きたいときはhline命令を使う。

¥hline ¥hline

とすれば，二重線になる。

表の最後は，¥end{tabular}で括る。これで表のできあがりといった寸法だ。

しかし，やはり表はセンタリングしてほしいし，本文との間に行間を多少とってほしい。表題などもつけてほしいし，文字だって本文よりちょっと小さいほうがよいと思うだろう。もっといえば，表目次を書くための参照ナンバーもほしいところ。

このように表を管理するためには，tabular環境だけでなく，ほかの命令も駆使する。同じ表にこれらのことをいろいろつけ加えてみよう(印字例2の1.2)。

リスト2の32行目を見てほしい。最初は{¥small……}でネストしている。これは単に表の文字列を小さくせよという指定。もっと小さくしたいのなら，footnotesizeでも使えばよいだろう。

そして表を管理するためのtable環境。table環境は例のようにtabular環境を囲むように設定をすればよい。この環境には，オプションがあって33行目のように[h]だと表は，TeXファイル中で表の記述が現れる場所に置かれる。ほかに[t]ならば表はページの先頭に，[b]なら表はページの終わりに置かれる。また，[p]を指定した場合，そのページは表だけのページになるので，表が大きいときに便利というわけだ。

この先の2行は，表を管理するうえで大きな意味をもつ。

ひとつは¥caption命令。フォーマットは，

¥caption[表目次用の文章]{表題}

といった感じ。これはtable環境のなかだけでしか有効でないので注意するように。表目次はいちばん最初のほうに，listoftables命令を書いておけば(リスト1，11行目)，整理されて書かれる。表題は，ちゃんと番号をつけて管理できる。この番号をつける命令が次のlabel命令で，記述は，

¥label{参照用の文字列}

と書くだけ。参照用の文字列は，例のように適当につければよい。まさにラベルをつけておくだけなので，文章中に，

¥ref{ラベル}

とするだけで，ここでつけた参照用の文字列と同じもので参照すれば，そこに表番号がつくという仕掛けである。こうすれば，表の場所を入れ替えたり，途中にセクションをつけ加えたりしても，ちゃんと文章中の参照は同じ場所を指すというわけだ。

次にネストするのは，center環境。センタリングの指定だ。ネストしたところをセンタリングするというだけの設定なので，表全体がセンタリングされるというわけ。

センタリングではなく，左寄せや右寄せをしたい場合はflushleft，flushright環境を使えばよい。center環境と同じように，オプションはまったくないため，

¥begin{flushleft}～¥end{flushleft}

でネストすれば，左寄せされる。

tabular環境でのテクニック

ここまでくれば，簡単な表はなんでも記述できるだろうが，もうちょっと欲を出して，美しい表を書いてみよう。

印字例2の表1を見ると，2つの項目は，CD-ROMユーザーに対しての項目だから，tabular環境の1つひとつのパラメータにわざわざ「CD-ROMを～」という項目を書くと，ひとつが長くなって美しくない。表にするときはちょっとしたテクニックを使って，わかりやすく書くと同時に，美しくも書きたい。

そういうわけで，印字例3の表2ではそのあたりに気をつけてみた。今回のtabular環境は|c| |c|c|c|c|と6項目をセンタリングし，ひと項目ごとに線を入れている。

ここで最初に出てくる新出命令はmulticolumn命令。これは，複数の項目の一部をまとめて書きたいときに使う命令だ。tabular環境では項目が6つに分かれているのに，表の1行目が4つに区切られていることに気がつけば，この意味はすぐにわかるだろう。フォーマットは，

¥multicolumn{ぶち抜く数}{そこでの環境}{中身}

といった感じだ。

印字例2

1 tabular 環境の簡単な使い方

1.1 単純に tabular 環境のみ使う

もっとも簡単な表です

CD-ROM ユーザー	CD-ROM は持ってない
50	3

直後に書いた文章はこうなります。

1.2 表を飾る

結果的に表を美しく書くには、私はこう設定します。

表 1: 身の回りの CD-ROM ユーザー (人)

CD-ROM ユーザー	CD-ROM を持ってない
50	3

と、まあ、こうなるのです。表の参照はこうすれば出来ます。表 1 を参照にして ...

リスト2

```
14: ¥newpage
15:
16: ¥section{tabular環境の簡単な使い方}
17:
18: ¥subsection{単純にtabular環境のみ使う}
19: もっとも簡単な表です
20:
21: ¥begin{tabular}{|cc|} ¥hline
22:   CD-ROMユーザー & CD-ROMを持ってない ¥¥ ¥hline ¥hline
23:       50       &        3              ¥¥ ¥hline
24: ¥end{tabular}
25:
26: 直後に書いた文章はこうなります。
27:
28: ¥subsection{表を飾る}
29:
30: 結果的に表を美しく書くには、私はこう設定します。
31:
32: {¥small                          % 文字を少し小さくする設定
33:  ¥begin{table}[h]
34:   ¥caption[身の回りのCD-ROMユーザーの数]{身の回りのCD-ROMユーザー(人)}       % 表に名前をつけた
35:   ¥label{tbl:cdrom}              % ラベルを張った
36:   ¥begin{center}                 % 表全体のセンタリング指定
37:    ¥begin{tabular}[b]{|cc|} ¥hline
38:     CD-ROMユーザー & CD-ROMを持ってない ¥¥ ¥hline ¥hline
39:         50       &        3              ¥¥ ¥hline
40:    ¥end{tabular}
41:   ¥end{center}
42:  ¥end{table}
43: }
44:
45: と、まあ、こうなるのです。表の参照はこうすればできます。
46:
47: 表¥ref{tbl:cdrom} を参照して ¥ldots
```

印字例3

2 tabular 環境ですこしテクを

美しく表記をするために、表2ではちょっとしたテクを使ってみた。

表 2: 私の周辺から無作為に 10 人選んだときの 68 ユーザの状況

主に友人と会う場所	CD-ROM を		MO を		本体実装
	所持	なし	所持	なし	メモリの平均
編集部関係	9	1	10	0	8MB
通信関係	10	0	10	0	8MB

(かなりいい加減な統計です)

リスト3

```
48:
49: ¥section{tabular環境ですこしテクを}
50:
51: 美しく表記をするために、表¥ref{tbl:cdrom2}ではちょっとしたテクを使ってみた。
52:
53: {¥small                        % 文字を少し小さくする設定
54: ¥begin{table}[h]
55:  ¥caption{私の周辺から無作為に10人選んだときの68ユーザの状況}        % 表に名前をつけた
56:  ¥label{tbl:cdrom2}      % ラベルを張った
57:   ¥begin{center}         % 表全体のセンタリング指定
58:    ¥begin{tabular}{|c||c|c|c|c|c|} ¥hline
59:      & ¥multicolumn{2}{c|}{CD-ROMを}
60:      & ¥multicolumn{2}{c|}{MOを}
61:      & 本体実装¥¥
62:      ¥cline{2-5}
63:     主に友人と会う場所 & 所持 & なし
64:      & 所持 & なし & メモリの平均 ¥¥ ¥hline ¥hline
65:     編集部関係 &  9 & 1 & 10 & 0 & 8MB ¥¥
66:     通信関係   & 10 & 0 & 10 & 0 & 8MB ¥¥ ¥hline
67:     ¥multicolumn{6}{r}{¥footnotesize (かなりいい加減な統計です)}
68:    ¥end{tabular}
69:  ¥end{center}
70: ¥end{table}
71: }
72:
```

まず{ぶち抜く数}というのは，tabular環境で指定した項目の現在の項目位置から何個つなげるかを意味する数。つまり，59行目では最初の&でいきなり第2項目に移る。それからmulticolumn命令で{2}が指定してあるから，第2，第3項目をつなげるという設定になるのだ。

次の{そこでの環境}というのはぶち抜いてしまったあとの環境を指定する命令。フォーマットはtabular環境のフォーマットと同じ。ここではセンタリングをし，縦線は最後にしか書いてほしくないので，c|とした。

最後は書く内容。これは見ればわかる。

このようにして，表2ではところどころでmulticolumnを利用しているが，こうしてぶち抜いたところには，とき折，横線がほしくなるときがある。それを実現するのが，¥cline命令だ。

¥cline{2-5}

というのは，第2～第5項目までに横線を引っ張るという命令。リスト3と表2を相互に見ればどこからどこまでの線かはすぐわかるだろう。

最後のmulticolumn命令は，表に下つきでちょこっとつけたいなと思ったので，この命令を利用して遊んでみた。まぁ，こんなこともtabular環境とmulticolumn命令を使えばできてしまうというサンプルだな。

tabbing環境

次にtabbing環境について触れてみようと思う。

tabbing環境にオプションはないのでそのままネストすれば，その部分がtabbing環境になる。リスト4(80行目)のように，最初の1行目は項目の内容と同時にTAB幅を設定するため，¥=で項目を区切る。

この場合ではtitle?という長さの文字列の区切りをTABとして設定をし，以降の内容は，その幅に合わせて出力される。

よって，title?の長さよりもtext?の長さが長い場合は，文字が重なってしまうので注意が必要だ。

なお，tabbing環境はcenter環境でもセンタリングできないので注意すること。下手にtable環境で処理しようにもcaption(表題)がとんでもないところに出てしまって，レイアウトが美しくない。

まあ，こんな感じで，tabbing環境はどうも使いにくいため，私はtabular環境のほうを使っている。

まとめ

まずはデフォルトの環境を駆使して，LaTeXしてみる。そういった目標で始めている。たとえば今回のtabular環境での二重線の間隔，表のサイズを固定する方法など，細かいこともあるが，これらは専門書を見てほしい。

TeXは綺麗なドキュメントを簡単に書くことができる。もちろん，細かい設定を行うことも可能だが，私個人は，XDTPが発売されたのだから，

・細かくレイアウトを考えて文章を書くならばXDTP
・簡単な文章を印刷するならシャーペン
・表や数式がたくさんあり，長めの文章を効率よく処理したいならTeX

と，使い分けている。

TeXは長い論文を書くのもとっても楽である。ref命令を使えば，このことがわかると思う。

さて来月は，環境についてもう少し補足をし，レイアウトに関することを書いていくつもりだ。

というわけで，今月はこれでおしまい。

※印字例（66%に縮小，プリンタはBJC-400Jを使用）

印字例4

3 tabbing 環境

tabbing 環境の例を示そう。

title1 title2 title3 title4
text1 text2 text3 text4
text5 text6 text7 text8

表の直後の文章はこの場所にくる。

リスト4

```
73: ¥newpage
74:
75: ¥section{tabbing環境}
76:
77: tabbing環境の例を示そう。
78:
79: ¥begin{tabbing}
80:  title1 ¥= title2 ¥= title3 ¥= title4 ¥¥
81:  text1  ¥> text2  ¥> text3  ¥> text4  ¥¥
82:  text5  ¥> text6  ¥> text7  ¥> text8
83: ¥end{tabbing}
84:
85: 表の直後の文章はこの場所にくる。
86:
87: ¥end{document}
```

Oh!X

LIVE in '94

X68000・Z-MUSIC
Ver.2.0用(SC-55対応)

幻想即興曲

Takahashi Toshiyuki 高橋 利之

X68000・Z-MUSIC
Ver.2.0用(SC-55対応)

「ヤマトタケル 魔空戦神異聞」より

風の少年

Morikami Akihito 森上 晶仁

X68000・Z-MUSIC
Ver.2.0用(SC-55対応)

「きまぐれ オレンジ☆ロード」より

オレンジ・ミステリー

Hayasaka Makoto 早坂 真

MIDIを使ったボリュームたっぷりの3作品です。入力は大変ですが，苦労のあとにはすてきな演奏が待っています。曲作りの参考にもなるでしょう。ジャンルもちょっと久しぶりに，ショパンのピアノ曲とテレビアニメからという選曲になりました。

ファンタジー

ではまず，クラシック曲からご紹介しましょう。ショパンの「幻想即興曲」です。たまにはこういう格式高いものもいいですね。クラシックは投稿数も少なめなので，採用には「狙い目」のジャンルです。が，発音上限のあるDTM音源，しかもMMLで，人間臭さの集大成であるこの手の曲を再現するのには相当高度なミュージックプログラミング技術が要求されます。

この「幻想即興曲」はショパンのピアノ曲のなかでも難しさはトップレベルです。右手と左手がわざとシンクロしていなかったり，1小節にやたら高速音符が散らばっていたりと，なんか半分テクニック誇示の曲という気がしないでもないですが，この曲が醸し出す，口では歌えないほどの高速ハーモニーはなんともいえない独特の美しさを持っています。このあたり，やはりショパンはピアノの天才ということを物語っています。

演奏のやり方によっては相当「熱い」曲にもなるこの「幻想即興曲」を高橋君は，極限まで人間臭さを演出して完成してくれました。リストは長めですが，まぁ聴いてください。ピアノを習っている人は録音して先生に「私の演奏です」とはったりかましてみましょう。顔が青ざめたら「うっそーん」とお尻を出してごまかしましょう。

日本ファンタジー

2曲目は，テレビアニメ「ヤマトタケル魔空戦神異聞」より「風の少年」です。見晴らしのいい高い丘から，目の前に広がる雄大な景色を見下ろしているような，さわやかでいて雄大な曲です。

聴きどころはクワイア，ピッコロのメロディユニゾンとこれに被るストリングスバックとが作り出すハーモニーです。オーケストラ用の音楽を作る場合，メロディをどんな風にするかという点も確かに重要ですが，違った種類の楽器同士が作り出すその場その場の「音の混合色」もきわめて重要です(音大の作曲科などでは，どの楽器とどの楽器を組み合わせるとこんな感じになる，などを勉強します)。そんな楽器同士が作り出す「色」はどんなものか，そういった視点でこの曲を聴いてみましょう。

きまぐれ オレンジ☆ロード

演奏にはGS音源が必要です。SC-55，SC-55mkIIどちらでも正常に演奏できます。

ヤマトの彼，再び

最後は，テレビアニメ「きまぐれ オレンジ☆ロード」のオープニングソングより「オレンジ・ミステリー」です。データ制作者は，本誌1994年4月号で超(長!?)大作データ「宇宙戦艦ヤマト」を発表した早坂君です。さすが，完成度はバッチリです。

作曲者はNOBODYなんですが，同じ

幻想即興曲

ヤマトタケル 魔空戦神異聞

NOBODYの曲でこれにそっくりなのがあります。「マリリンマリリーン」とかいうやつです。興味がある方はそちらもどーぞ。

データは打ち込んだところまでを全パートが演奏できるような構成になっており，さらに歌詞との対応も記述してあるので，飽きずに入力できると思います。使用楽器はSC-55。SC-55mkIIやSC-88でも演奏はできますが，ニュアンスが多少違ってきます(わずかですが)。MT-32バリエーション音色を使用しているためSC-33などでは演奏がおかしくなってしまいます。(Z.N)

リスト1　幻想即興曲

```
  1: .COMMENT  Fantasie Impromptu(幻想即興曲) F.Chopin  By バビィ
  2:
  3: (i)
  4: (B1)
  5:
  6: .ROLAND_EXCLUSIVE $10,$42={$40,$00,$7F,$00}
  7: .SC55_V_RESERVE $10={24,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
  8:
  9: (m1, 8000)(aMIDI1,1)
 10: (m2, 6000)(aMIDI1,2)
 11: (m3,11000)(aMIDI1,3)
 12:
 13: / Piano(右手)
 14: (t1)
 15:   @I$41,$10,$42Y$42,127Y$43.40I0
 16:   @E80@p64@1q8v16r32
 17: /---1.1
 18:   [K.SIGN +c,+d,+f,+g]
 19:   r1
 20:   r1
 21:   r1
 22:   r1
 23:   L16@u48ro4gagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
 24: /---1.2
 25:   z66,70,72,76,78,80,82,78r>gagg!g<cedcd@u76c>b+<ce@u66g
 26:   z76,78,80,82,86,86,88,90,90,86,84,82,80,78,76r>a<cdfa<cdba
gfedfc
 27:   @u74>b+<d>agfaedfc>b+<d>ag@u70b@u66a&
 28: /---1.3
 29:   a@u48gagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
 30:   z66,70,72,76,78,80,82,78,80,82,84,86,88,90,92r>ga+gg!g<ced
cdc>b+<ceg
 31:   z88,90,92,94,90,84,78,72,80,78,76,74,72,70,68,66dedd!dba+g
g!<edc>ba+gg!
 32: /---1.4
 33:   z65,65,64,64,63,63,62,62,61,61,60,60,59,59,60,62a+gbd!edg>
a+<c>b<d>g!a+gg!g
 34:   z61,62,63,65,66,67,73,74,75,81,82,83r<g>b+<crf>b+<cre+>b+<
crf>b+<c
 35:   z88,78,77,77,86,75,74,72,80,68,66,64,72,60,58,56Y$43,0>cY$
43,40<c>faY$43,0dY$43,40<d>faY$43,0eY$43,40<e>gbY$43,0gY$43,40<g>b<e
 36: /---1.5
 37:   z70,62,63,63,72,65,66,67,76,69,70,71,80,74,76,76Y$43,0>gY$
43,40<g>b+<cY$43,0>fY$43,40<f>b+<cY$43,0>e+Y$43,40<e+>b+<cY$43,0>fY$
43,40<f>b+<c
 38:   z86,78,77,77,84,74,72,70,76,66,64,62,68,58,56,54Y$43,0>e+Y
$43,40<e+>b<dY$43,0>fY$43,40<f>b<dY$43,0>aY$43,40<a>b<eY$43,0>gY$43,
40<g>b<e
 39:   @u48>grb+<c>frb+<c>e+rb+<c>frb+<c
 40: /---2.1
 41:   z49,60,51,52,53,64,55,56,57,68,59,60,61,72,63,64>cY$43,0<c
Y$43,40>fadY$43,0<dY$43,40>faeY$43,0<eY$43,40>gbgY$43,0<gY$43,40>b<e
 42:   z65,76,67,68,69,80,71,72,76,86,76,76,76,86,76,76>gY$43,0<g
Y$43,40>b+<c>fY$43,0<fY$43,40>b+<cdY$43,0<dY$43,40>facY$43,0<cY$43,4
0>fa
 43:   z76,86,76,76,80,90,80,80,80,90,80,80,76,86,76,76cY$43,0<cY
$43,40>df>b+Y$43,0<b+Y$43,40df>b+Y$43,0<b+Y$43,40df>b+Y$43,0<b+Y$43,
40df
 44: /---2.2
 45:   z78,86,74,72,70,78,66,64,62,70,58,56,54,62,50,48c!Y$43,0<c
!Y$43,40>df>bY$43,0<bY$43,40df>bY$43,0<bY$43,40df>a+Y$43,0<a+Y$43,40
df
 46:   z47,56,45,44,43,52,41,40,39,48,37,36,35,44,33,32>a+Y$43,0<
a+Y$43,40df>aY$43,0<aY$43,40df>aY$43,0<aY$43,40df>aY$43,0<aY$43,40df
 47:   z32,42,32,32,32,42,32,32,32,42,32,32,32,42,32,32c!Y$43,0<c
!Y$43,40>df>bY$43,0<bY$43,40df>bY$43,0<bY$43,40df>a+Y$43,0<a+Y$43,40
df
 48: /---2.3
 49:   z32,42,32,32,31,41,30,30,29,39,28,28,27,37,26,26>a+<a+df>a
<adf>a<adf>a<gdf
 50:   @u48r>gagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
 51:   z66,70,72,76,78,80,82,78r>gagg!g<cedcd@u76c>b+<ce@u66g
 52: /---2.4
 53:   z76,78,80,82,84,86,88,90,88,86,84,82,80,78,76r>a<cdfa<cdba
gfedfc
 54:   @u74>b+<d>agfaedfc>b+<d>agb@u70a&
 55:   a@u48gagz51,52,53,54,56,58,60,62,64,66,68,70g!g<cedcdc>b+<
ceg
 56: /---2.5
 57:   z72,74,76,78,80,82,84,86,72,70,68,66,64,62,56,50ededd!dfaf
e+fe+ee+g<c
 58:   z48,49,50,51,52,53,54,55,56,57,58,59,60,61,62r>cd!c>b+<cfa
fe+fe+ee+g<c
 59:   z63,64,65,66,68,70,72,74,75,76,77,78,80,82,84,86>gfgfe+fa<
c>agagg!gb+<d
 60: /---2.6
 61:   @u88>gagg!gY$43,30@u96<e@u90dd!cc!>ba+agg!f
 62:   efedeY$43,20@u96<e@u92dd!Y$43,10cc!>ba+agg!f
 63:   grY$43,0@u96<gg!fe+edd!cc!>ba+agg!
 64: /---3.1
 65:   fe+edd!cc!>ba+agg!fed@u100c
 66:   z110,106,96,96,96,106,96,96,96,106>g8r8<<<ag<e>efe<c>cd@u9
6cg>g
 67:   ag<e>efe<c>cdcg>gag<e>e
 68: /---3.2
 69:   fe<c>cdcg>g<dcg>g<dca>a
 70:   <dcg>g<dcg!>g!z94,92,90,88,86,84,82,80Y$43,20<dcg>gY$43,40
<d>b+Y$43,60<g>g
 71: /---3.3
 72:   [K.SIGN -d,-e,-g,-a,-b]
 73:   Y$43,40r1
 74:   r1
 75:   o4@u48a2@u52b*3<c*3>b*18a8<d8e8
 76:   f2@u48a2
 77: /---3.4
 78:   L4@u52gfef8.@u48d16
 79:   z46,64>a2b2&
 80:   z64,56,56,56,60,64b2<c-*4d*4c-*16>b8<e8f8
 81:   z60,64,60,54gfef
 82: /---3.5
 83:   @u48d4.Y$43,80@u40c32d32e32d32f.Y$43,40e8
 84:   @u46e1
 85:   z50,52,52,52,60,64,68>a2Y$43,60b*3<c*3>b*18Y$43,40a8<d8e8
 86:   @u72f2a2
 87: /---4.1
 88:   z76,70,64,58,58,58,58,52,48gfe8Y$43,60f32e32d32e32f8.Y$43,
40d16
 89:   @u40>a2@u48b2&
 90:   b2Y$43,60@u52<c-*3d*3c-*18Y$43,40z60,56,60>b8<e8f8
 91: /---4.2
 92:   z56,52,48,44gfef
 93:   z40,40,44,48,44,40,40d.Y$43,80>g!32a32b32a32<f.Y$43,40e8
 94:   z36,32,50e2dr8a8
 95:   z60,68,68,68,72,74,76>a2Y$43,60b*3<c*3>b*30Y$43,40a!16b8.<
c16
 96: /---4.3
 97:   L8@u80>ar@u60<<c4&Y$43,80@u38{c&>baf-d}8>b*56@u42<a12
 98:   Y$43,40z36,46,48,44,42,40>e4...Y$43,80<c32>b8.Y$43,40a!16b
.<e16
 99:   L4z38,48>arb@u38Y$43,80{<dc>b}
100:   Y$43,40z34,40,40,40,42,44,46a2Y$43,80b*2<c*2>b*20Y$43,40a8
<d8e8
101: /---4.4
102:   z50,56f2a2
103:   z54,50,46,44,44,42,42,40,38gfe8Y$43,80f32e32d32e32f8Y$43,4
0d8
104:   z32,64>a2b2&
105:   z64,62,64,66,72,74,76b2Y$43,60<c-*2d*2c-*20Y$43,40>b8<e8f8
106: /---4.5
107:   z80,78,76,74gfef
108:   z70,62,60,58,56,56,54d.Y$43,60>g!32a32b32a32<f.Y$43,40e8
109:   z52,48,56e2dr8a8
110: /---5.1
111:   z80,80,80,80,88,92,96>a2Y$43,60b*2<c*2>b*32Y$43,40a!16b8.<
c16
112:   L8z80,50>ar<<c4&Y$43,80@u32{c&>baf-d}8>b*56@u28<a12
113:   Y$43,40@u32>e4...Y$43,80<c32>b8.Y$43,40a!16b.<e16
114: /---5.2
115:   L4>arz32,32,30,28bY$43,80{<dc>b}
116:   Y$43,40z32,50,50,50,54,58,62a2Y$43,60b*3<c*3>b*18a8<d8e8
117:   z66,72f2a2
118: /---5.3
119:   z76,72,64,62,60,58,56,52,48gfe8Y$43,80f32e32d32e32f8Y$43.4
0d8
120:   z42,38>a2b2&
121:   b2Y$43,80z40,40,40,42,44,46<c-*2d*2c-*20Y$43,40>b8<e8f8
122: /---5.4
123:   z50,54,50,46gfef
124:   z40,38,36,34,32d.Y$43,80>g!32a32b32a32@u32<f.Y$43,40e8
125:   e1
126: /---5.5
127:   [K.SIGN +c,+d,+f,+g]
128:   L16@u48o4rgagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
129:   z66,70,72,76,78,80,82,78r>gagg!g<ced@u76cdc>b+<ce@u66g
130: /---6.1
131:   z76,78,80,82,86,86,88,90,90,86,84,82,80,78,76r>a<cdfa<cdba
gfedfc
132:   @u74>b+<d>agfaedfc>b+<d>ag@u70b@u66a&
133: /---6.2
134:   a@u48gagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
135:   z66,70,72,76,78,80,82,78,80,82,84,86,88,90,92r>ga+gg!g<ced
cdc>b+<ceg
136: /---6.3
137:   z88,90,92,94,90,84,78,72,80,78,76,74,72,70,68,66dedd!dba+g
g!<edc>ba+gg!
138:   z65,65,64,64,63,63,62,62,61,61,60,60,59,59,60,62a+gbd!edg>
a+<c>b<d>g!a+gg!g
139: /---6.4
140:   z61,62,63,65,66,67,69,70,71,74,76,78r<g>b+<crf>b+<cre+>b+<
crf>b+<c
141:   z84,78,77,77,82,75,74,72,80,68,66,64,72,60,58,56Y$43,0>cY$
43,40<c>faY$43,0dY$43,40<d>faY$43,0eY$43,40<e>gbY$43,0gY$43,40<g>b<e
142: /---6.5
143:   z70,62,63,63,72,65,66,67,76,69,70,71,80,74,76,76Y$43,0>gY$
43,40<g>b+<cY$43,0>fY$43,40<f>b+<cY$43,0>e+Y$43,40<e+>b+<cY$43,0>fY$
43,40<f>b+<c
144:   z86,78,77,77,84,74,72,70,76,66,64,62,68,58,56,54Y$43,0>e+Y
$43,40<e+>b<dY$43,0>fY$43,40<f>b<dY$43,0>aY$43,40<a>b<eY$43,0>gY$43,
40<g>b<e
145: /---7.1
146:   @u48>grb+<c>frb+<c>e+rb+<c>frb+<c
147:   z49,60,51,52,53,64,55,56,57,68,59,60,61,72,63,64>cY$43,0<c
Y$43,40>fadY$43,0<dY$43,40>faeY$43,0<eY$43,40>gbgY$43,0<gY$43,40>b<e
148: /---7.2
149:   z65,76,67,68,69,80,71,72,76,86,76,76,76,86,76,76>gY$43,0<g
Y$43,40>b+<c>fY$43,0<fY$43,40>b+<cdY$43,0<dY$43,40>facY$43,0<cY$43,0
>fa
150:   z76,86,76,76,80,90,80,80,80,90,80,80,76,86,76,76cY$43,0<cY
$43,40>df>b+Y$43,0<b+Y$43,40df>b+Y$43,0<b+Y$43,40df>b+Y$43,0<b+Y$43,
40df
151:   z78,86,74,72,70,78,66,64,62,70,58,56,54,62,50,48c!Y$43,0<c
!Y$43,40>df>bY$43,0<bY$43,40df>bY$43,0<bY$43,40df>a+Y$43,0<a+Y$43,40
df
152: /---7.3
```


▶３Ｄモノに興味を持っているため，ハードコア３Ｄエクスタシーに注目しているのですが……やはり難しい。　坂井　純一(24)神奈川県

```
   153:   z47,56,45,44,43,52,41,40,39,48,37,36,35,44,33,32>a+Y$43,0<
a+Y$43,40df>aY$43,0<aY$43,40df>aY$43,0<aY$43,40df>aY$43,0<aY$43,40df
   154:   z32,42,32,32,32,42,32,32,32,42,32,32,32,42,32,32c!Y$43,0<c
!Y$43,40>df>bY$43,0<bY$43,40df>bY$43,0<bY$43,40df>a+Y$43,0<a+Y$43,40
df
   155: /---7.4
   156:   z32,42,32,32,31,41,30,30,29,39,28,28,27,37,26,26>a+<a+df>a
<adf>a<adf>a<gdf
   157:   @u48r>gagz50,52,54,56,52g!g<ced@u48cdc>b+<ce@u44g
   158:   z66,70,72,76,78,80,82,78r>gagg!g<cedcd@u76c>b+<ce@u66g
   159: /---7.5
   160:   z76,78,80,82,84,86,88,90,88,86,84,82,80,78,76r>a<cdfa<cdba
gfedfc
   161:   @u74>b+<d>agfaedfc>b+<d>agb@u70a&
   162: /---8.1
   163:   a@u48gagz51,52,53,54,56,58,60,62,64,66,68,70g!g<cedcdc>b+<
ceg
   164:   z72,74,76,78,80,82,84,86,72,70,68,66,64,62,56,50ededd!dfaf
e+fe+ee+g<c
   165:   z48,49,50,51,52,53,54,55,56,57,58,59,60,61,62r>cd!c>b+<cfa
fe+fe+ee+g<c
   166: /---8.2
   167:   z63,64,65,66,68,70,72,74,75,76,77,78,80,82,84,86>gfgfe+fa<
c>agagg!gb+<d
   168:   @u88>gagg!gY$43,20@u96<e@u90dd!cc!>ba+agg!f
   169: /---8.3
   170:   efedeY$43,20@u96<e@u92dd!Y$43,10cc!>ba+agg!f
   171:   grY$43,0@u96<gg!fe+edd!cc!>ba+agg!
   172: /---8.4
   173:   fe+edd!cc!>ba+agg!fed@u100c
   174:   z110,106,96,96,96,106,96,96,96,106>g8r8<<<ag<e>efe<c>cd@u9
6cg>g
   175: /---8.5
   176:   ag<e>efe<c>cdcg>gag<e>e
   177:   fe<c>cdcg>g<dcg>g<dca>a
   178:   <dcg>g<dcg!>g!z94,92,90,88,86,84,82,80Y$43,20<dcg>gY$43,40
<d>b+Y$43,60<g>g
   179: /---9.1
   180:   z78,76,74,80<dcg>gY$43,0@u96<agreagreagrd
   181:   z96,96,106,96,96,96,100,96,84,80,90,74,74,74,84,74ag<d>dag
<c>cedg>g<edg>g
   182:   z74,74,84,80,96,96,106,96,96,96,106,96,96,96,106,96<dcg>g<
ag<e>eag<e>eag<d>d
   183: /---9.2
   184:   z96,96,106,96,96,96,100,96,84,80,90,74,74,74,84,74ag<d>dag
<c>cedg>g<edg>g
   185:   z74,74,84,80,96,96,106,96,96,96,106,96,96,96,106,96<dcg>g<
ag<e>efe<c>cag<e>e
   186:   z96,96,106,96,96,96,106,96,96,96,106,96,96,96,106,96dcg>g<
ag<e>efe<c>cag<e>e
   187: /---9.3
   188:   z95,94,93,92,91,90,89,88,87,86,85,84,83,82,81,80fe<c>cag<c
>cfe<c>cag<c>c
   189:   Y$43,10z79,78,77,76,75,74,73,72,71,70,69,68,67,66,65,64fe<
c>cag<c>cfe<c>cag<c>c
   190:   Y$43,20z63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,48dcg
>g<dcg>g<dcg>g<dcg>g
   191: /---9.4
   192:   Y$43,30z47,46,45,44,43,42,41,40,39,38,37,36,35,34,33<dcg>g
<dcg>g<dcg>g<dcg@u32>g
   193:   Y$43,40<dcg>g<dcg>g<dcg>g<dcg>g
   194:   <dcg>g<dcg>g<dcg>g<dcg>g
   195: /---9.5
   196:   <dcg>g<dcg>g<dcg>g<dcg>g
   197:   <dcg>g<dcg>g<dcg>g<dcg>g
   198:   <dcg>g<dcg>g<dcg>g<dcg>g
   199: /---9.6
   200:   @u31<dcg>g<dcg>g<dcg>g<dcg>g
   201:   @u30<dcg>e+<dcg>e+<dcg>e+<dcg>e+
   202:   L8@u29<e+dg>f<e+dg>f<e+dg>f<e+dg>f
   203:   @u28o2c*0r'g2..<fgb+<d',6&
   204:   c*0r16g*0r16<e+*0rg*0r4<c2
   205:
   206: / Piano(左手)
   207: (t2)
   208:   @I$41,$10,$42Y$42,127I0
   209:   @E80@p64@1q8v16r32
   210: /---1.1
   211:   [K.SIGN +c,+d,+f,+g]
   212:   Y$43,0@u100o3'g1>g'&
   213:   'g1>g'
   214:   L12@u80'c>c'g<cec>gY$43,20cg<cz78,76,74ec>g
   215:   Y$43,40z72,70,68,66,64,62,60,58,56,54,52,50cg<cec>gcg<cec>
g
   216:   @u48cg<cec>g@u52eg<cec>g
   217: /---1.2
   218:   @u60cg<cec>g@u64eg<cec>g
   219:   @u70da<cfc>a@u74f<cdadc
   220:   @u68>>g<dfb+fd@u64>g<dfb+fd
   221: /---1.3
   222:   @u48cg<cec>g@u52eg<cec>g
   223:   @u60cga+<e>a+g@u64ega+<c>a+g
   224:   @u68dgb<d>bg@u72da+<cg!c>a+
   225: /---1.4
   226:   @u68>g<dgbgd@u64>g<dgbgd
   227:   @u60>a<cfafc@u64>a<cfafc
   228:   @u68>b<fabaf@u64>eb<ege>b
   229: /---1.5
   230:   @u68a<cfafc@u72>a<cfafc
   231:   @u64>b<dabad@u58>eb<ege>b
   232:   @u48a<cfafc>a<cfafc
   233: /---2.1
   234:   @u52>b<fabaf@u56>eb<ege>b
   235:   @u60a<cfafc@u64>a<dfafd
   236:   @u68>g<dfb+fd@u72>g<dfb+fd
   237: /---2.2
   238:   @u76>a<dfbfd@u68>a+<dfa+fd
   239:   @u48>b+<dfafd@u40>b+<dfafd
   240:   @u32>a<dfbfd@u36>a+<dfa+fd
   241: /---2.3
   242:   @u32>b<dfafd@u28>b+<dfgfd
   243:   @u48cg<cec>g@u52eg<cec>g
   244:   @u60ce<cec>g@u64eg<cec>g
   245: /---2.4
   246:   @u70da<cfc>a@u74f<cdadc
   247:   @u68>>g<dfb+fd@u64>g<dfb+fd
   248:   @u48cg<cec>g@u52eg<cec>g
   249: /---2.5
   250:   @u60f<cdadc@u64>g<ce+be+c
   251:   @u60>a<cfafc@u64>g<ce+be+c
   252:   @u68>a<cfafc>>b+<fg<d>gf
   253: /---2.6
   254:   @u72cg<cY$43,30ec>gdgb+<f>b+g
   255:   @u76eg<cY$43,20ec>gY$43,10@u80f<cdadc
   256:   Y$43,0@u96'c8eg>g'&r32r*66@u102'c2ea>g!'&
   257: /---3.1
   258:   'c2.ea>g!'r4
   259:   @u110'>>g8>g'r8L4@u96'e<e''c<c'>'g<g'
   260:   'e<e''c<c'>'g<g''e<e'
   261: /---3.2
   262:   'c<c''g>g''g>g''a>a'
   263:   'g>g''g!>g!''g2>g'
   264: /---3.3
   265:   [K.SIGN -d,-e,-g,-a,-b]
   266:   z100,92,88,84,80,76,72,68,64,60,56,52L12o2dY$43,40a<dfa<df
d>afd>a
   267:   z48,46,44,42,40,38,37,36,35,34,33,32da<dfa<dfd>afd>a&
   268:   @u32a<a<cec>ada<cgc>a
   269:   da<dfd>ada<frf>a
   270: /---3.4
   271:   @u40r<e>ar<d>ar<c>araf
   272:   @u32>a<ea<c>ae@u48>g<gb<d>bg
   273:   @u52>b<fb<d!>bf@u56>b<b<d!ad!>b
   274:   @u52e<g>b@u48f<f>a!@u44g<e>b@u40a<gc
   275: /---3.5
   276:   @u36>df<dfd>a@u32eg!<ded>g!
   277:   @u40>a<ea<c>ae@u36>a<egbge
   278:   @u32>a<ea<c>ae@u36>a<a<cgc>a
   279:   @u40da<dfd>ad@u44a<frf>a
   280: /---4.1
   281:   z54,52,50,48,46,44,42,40r<e>ar<d>ar<c>araf
   282:   @u32>a<ea<c>ae@u40>g<gb<d>bg
   283:   @u44>b<fb<d!>bf>b<b<d!ad!>b
   284: /---4.2
   285:   @u48e<g>b@u44f<f>a!@u40g<e>b@u36a<gc
   286:   @u32>da<dfd>ada<cgc>a
   287:   da<cgc>ada<dfd>a
   288:   @u52>a<ea<c>ae>a<eg!@u56<d>g!e
   289: /---4.3
   290:   @u60>a<ea@u36<c>ae@u32da<df-d>a
   291:   >e<ea<c>ae>e<eg!<d>g!e
   292:   >a<ea<c>ae>g<gb<d>bg
   293:   >a<a<cec>aea<cgc>a
   294: /---4.4
   295:   da<dfd>ada<frf>a
   296:   r<e>ar<d>ar<c>araf
   297:   >a<ea<c>ae@u52>g<gb@u56<d>bg
   298:   @u60>b<fb<d!>bf@u64>b<b<d!ad!>b
   299: /---4.5
   300:   @u60e<g>b@u56f<f>a!@u52g<e>b@u48a<gc
   301:   @u44>da<dfd>a@u40da<cgc>a
   302:   @u36da<cgc>a@u32da<dfd>a
   303: /---5.1
   304:   @u56>a<ea<c>ae@u60>a<eg!@u64<d>g!e
   305:   @u68>a<eaz52,46,40<c>ae@u32da<df-d>a
   306:   >e<ea<c>ae>e<eg!<d>g!e
   307: /---5.2
   308:   >a<ea@u30<c>ae@u28>g<gb<d>bg
   309:   @u40>a<ea@u44<c>ae@u48>a<a<c@u52gc>a
   310:   @u56da<d@u60fd>a@u64da<f@u68rf>a
   311: /---5.3
   312:   r<e>a@u60r<d>a@u48r<c>a@u40raf
   313:   @u36>a<ea<c>ae>g<gb<d>bg
   314:   >b<fb<d!>bf>b<b<d!ad!>b
   315: /---5.4
   316:   e<g>bf<f>a!@u32g<e>ba<gc
   317:   >da<dfd>ada<cgc>a
   318:   @u28da<cgc>ada<cgc>a
   319: /---5.5
   320:   [K.SIGN +c,+d,+f,+g]
   321:   @u48o3cg<cec>g@u52eg<cec>g
   322:   @u60cg<cec>g@u64eg<cec>g
   323: /---6.1
   324:   @u70da<cfc>a@u74f<cdadc
   325:   @u68>>g<dfb+fd@u64>g<dfb+fd
   326: /---6.2
   327:   @u48cg<cec>g@u52eg<cec>g
   328:   @u60cga+<e>a+g@u64ega+<c>a+g
   329: /---6.3
   330:   @u68dgb<d>bg@u72da+<cg!c>a+
   331:   @u68>g<dgbgd@u64>g<dgbgd
   332: /---6.4
   333:   @u60>a<cfafc@u64>a<cfafc
   334:   @u68>b<fabaf@u64>eb<ege>b
   335: /---6.5
   336:   @u68a<cfafc@u72>a<cfafc
   337:   @u64>b<dabad@u58>eb<ege>b
   338: /---7.1
   339:   @u48a<cfafc>a<cfafc
   340:   @u52>b<fabaf@u56>eb<ege>b
   341: /---7.2
   342:   @u60a<cfafc@u64>a<dfafd
   343:   @u68>g<dfb+fd@u72>g<dfb+fd
   344:   @u76>a<dfbfd@u68>a+<dfa+fd
   345: /---7.3
   346:   @u48>b+<dfafd@u40>b+<dfafd
   347:   @u32>a<dfbfd@u36>a+<dfa+fd
   348: /---7.4
   349:   @u32>b<dfafd@u28>b+<dfgfd
   350:   @u48cg<cec>g@u54eg<cec>g
   351:   @u64ce<cec>g@u68eg<cec>g
   352: /---7.5
   353:   @u72da<cfc>a@u76f<cdadc
   354:   @u78>>g<df@u74b+fd@u72>g<df@u68b+fd
   355: /---8.1
   356:   @u48cg<cec>g@u52eg<cec>g
   357:   @u60f<cdadc@u64>g<ce+be+c
   358:   @u60>a<cfafc@u64>g<ce+be+c
```

▶PCMのFILTERを変更してみたところ，思った以上によい音がするではないか！ まだまだX68000には未知の能力が隠されているのかもしれない。　佐村 和亮(20)福岡県

```
 359: /---8.2
 360:    @u68>a<cfafc>>b+<fg<d>gf
 361:    @u72cg<cY$43,30ec>g@u76dgb+<f>b+g
 362: /---8.3
 363:    @u80eg<cY$43,20ec>gY$43,10@u84f<cdadc
 364:    Y$43,0@u96'c8eg>g'&r32r*66@u102'c2ea>g!'&
 365: /---8.4
 366:    'c2.ea>g!',0r4
 367:    @u110'>>g8>g'r8L4@u96'e<e''c<c'>'g<g'
 368: /---8.5
 369:    'e<e''c<c'>'g<g''e<e'
 370:    'c<c''g>g''g>g''a>a'
 371:    'g>g'Y$43,20'g!>g!'Y$43,40'g2>g'
 372: /---9.1
 373:    L8z78,74c<e@u96c>gc<f>b+g
 374:    c<ec>g@u74c<f>b+g
 375:    c<e@u96c>gc<f>b+g
 376: /---9.2
 377:    c<ec>g@u74c<f>b+g
 378:    c<e@u96c>gc<gc>g
 379:    c<ec>gc<gc>g
 380: /---9.3
 381:    z95,94,93,92,91,..,39,88c<ec>gc<ec>g
 382:    z87,86,85,84,83,82,81,80c<ec>gc<ec>g
 383:    L4cr2.
 384: /---9.4
 385:    r1
 386:    @u60'c1<g'
 387:    z64,72,68,60a+g<cd
 388: /---9.5
 389:    @u58e+1
 390:    @u56g1
 391:    z54,52f2e+2
 392: /---9.6
 393:    z50,48,46d2e+c
 394:    @u44>g1
 395:    @u40'g*384>g'
 396:    r1
 397:    r1
 398:
 399: / Tempo & Ped
 400: (t3)
 401:    @I$41,$10,$42Y$42,127Y$43,20I0
 402:    @E80@p64@1q8v16L16r32
 403: /---1.1
 404:    [K.SIGN +c,+d,+f,+g]
 405:    @D1t168rrrrrrrrrrrrrrr
 406:    rrrrrrrrrrrrrrr
 407:    @D0@D1t30rt40rt70rt100rt130rt150rt168rrrrrrrrr
 408:    rrrrrrrrrrrrrrr
 409:    @D0@D1rt190rrrt168rrrr@D0@D1rrrrrrt100rt70r
 410: /---1.2
 411:    @D0@D1t168rt190rrrt168rrrr@D0@D1rrrrrrt120rt100r
 412:    t168@D0@D1rrrrrrrr@D0@D1rrrrrrrr
 413:    @D0@D1rrrrrrrr@D0@D1rrrrr@D0@D1rrr
 414: /---1.3
 415:    @D0@D1t90rt190rrrt168rrrr@D0@D1rrrrrrt100rt70r
 416:    @D0@D1t168rt190rrrt168rrrr@D0@D1rrrrrrrt145r
 417:    @D0@D1t168rrrrt140rt130rt120rt110r@D0@D1rt120rrrt123rt126r
t129t132rt135r
 418: /---1.4
 419:    @D0@D1t138rt142rt146rt150rt154rt158rt162rt166rt168@D0@D1rr
rrrt120rt100rt80r
 420:    @D0@D1t100@u72o4g&t105g&t110g&gt115@u78f&t120f&t125f&t130f
@D0@D1t135@u86e+&e+&e+&e+t140@u94f&f&f&f
 421:    @D0@D1t145rrrrt130rt125rt120rt110r@D0@D1t95rt100rt105rt110
rt115rt120rrr
 422: /---1.5
 423:    @D0@D1t125rt130rt135rt140rt145rt150rt155rt160r@D0@D1rrrrrr
rr
 424:    @D0@D1rrrrt150rt145rt125rt110r@D0@D1t100rt95rt90rt85rt80rr
rr
 425:    @D0@D1@u58o5t90rt95g&t100g&t105gt110rt115f&t120f&t125f@D0@
D1t130rt135e+&t140e+&t145e+t150rf&f&f
 426: /---2.1
 427:    @D0@D1t155rrrrrrt130rt100r@D0@D1t80rt90rt100rt110rt120rt13
0rt140rt150r
 428:    @D0@D1t155rrrrt160rrrt130r@D0@D1t110rt115rt120rt125rt130rt
132rt135rt137r
 429:    @D0@D1t140rt142rt145rt147rt150rt155rt160rt165r@D0@D1t170rr
rrrrr
 430: /---2.2
 431:    @D0@D1t160rrt157rrt155rrt152rr@D0@D1t150rrt147rrt145rrt142
rr
 432:    @D0@D1t140rrt137rrt135rrt132rr@D0@D1t130rt125rt120rt110rt1
00rt90rt75rt60r
 433:    @D0@D1t80rt85rt90rt95rt100rt105rt110rt115r@D0@D1t120rt125r
t130rt135rt140rt137rt135rt132r
 434: /---2.3
 435:    @D0@D1t130rt125rt120rt115rt110rt107rt105rt102r@D0@D1t100rt
97rt95rt90rt85rt80rt75rt70r
 436:    @D0@D1t100rt190rrrt170rrrr@D0@D1rrrrrrt100rt70r
 437:    t170rt190rrrt170rrrrrrrrrrt120rt100r
 438: /---2.4
 439:    t170rrrrrrrrrrrrrrr
 440:    rrrrrrrrrrrrrrt140r
 441:    t120rt190rrrt170rrrrrrrrrrr
 442: /---2.5
 443:    rrrrrrrrrrrrrrr
 444:    t100rt190rrrt170rrrrrrrrrrr
 445:    rrrrrrrrrrrrrrr
 446: /---2.6
 447:    t190rrrrrt170rrrrrrrrrr
 448:    t190rrrrrt170rrrrrrrrrr
 449:    @D0@D1rr@D0t140rt150rt160rt170rrr@D1rrrr@D0rrrr
 450: /---3.1
 451:    rrrrrrrrrrt160rt140rt120rt100rt80r
 452:    @D1t50rrt170rrrrrrrrrrrrr
 453:    rrrrrrrrrrrrrrr
 454: /---3.2
 455:    rrrrt160rt155rt150rt145r@D0@D1t140rt135rt130rt125r@D0@D1t1
20rt110rt100rt90r
 456:    @D0@D1rrrr@D0@D1t85rrrr@D0@D1t80rrt75rr@D0@D1t70rrt65rr
 457: /---3.3
 458:    [K.SIGN -d,-e,-g,-a,-b]
 459:    @D0@D1t20rt40rt50rt60rt70rt80rrrrrrrrrt65rt50r
 460:    t25rt40rt50rt60rt70rt80rrrrrrrt70rt60rt50rt40r
 461:    @D0@D1t60@u32o2d&drrrrrrrrrrrrr
 462:    @D0@D1rrrrrrrt50r@D0@D1t60rrrr@u40o4a&a&t50a&t40a
 463: /---3.4
 464:    @D0@D1t60@u40g&g&g&g@D0@D1f&f&f&f@D0@D1e&e&e&e@D0@D1t55d&t
50d&t45d&t35d
 465:    @D0@D1rt45rt60rrrrrr@D0@D1rt62rt64rt66rt68rt70rrr
 466:    @D0@D1rrrrrrrr@D0@D1rrrrt72rt75rt77rt80r
 467:    @D0@D1rrrr@D0@D1rrrr@D0@D1t77rt75rt72rt70r@D0@D1t65rt60rt5
5rt50r
 468: /---3.5
 469:    @D0@D1t60rt62rt65rt67rt60rt55rt50rr@D0@D1t65rt67rt70rt72rt
65rt60rt55rt50r
 470:    @D0@D1t65rrrrrrrr@D0@D1rrrrt60rt50rt40rt30r
 471:    @D0@D1rt60rrrrrrt50rt60rrrrt70rt65rt60rr
 472:    @D0@D1t65rrrrt60rt57rt55rt50r@D0@D1rt55rrt50r@u54t35a&t45a
&t50a&t65a
 473: /---4.1
 474:    @D0@D1t65@u54g&g&g&g@D0@D1@u50f&f&f&f@D0@D1t60@u46e&e&e&e@
D0@D1t57@u42d&t55d&t52d&t50d
 475:    @D0@D1t60rrrrt55rt50rt45rt40r@D0@D1t50rt55rt60rrrrrr
 476:    @D0@D1rrrrt65rrrt50r@D0@D1t60rt65rrrt70rrrt65r
 477: /---4.2
 478:    @D0@D1rrt62rr@D0@D1t60rrt57rr@D0@D1t55rrt52rr@D0@D1t50rrrt
45r
 479:    @D0@D1t50rt55rt60rrrrrt45r@D0@D1t50rt55rt60rrrrrt45r
 480:    @D0@D1t50rt55rt60rrrrrt45r@D0@D1t50rt60rrrrrt40rt30r
 481:    @D0@D1t50rt60rt70rrrrrt45r@D0@D1t50rt60rt70rrrrrt45r
 482: /---4.3
 483:    @D0@D1t50rrt47rrt45rrt42rr@D0@D1t55rrrrt50rt47rt45rt30r
 484:    @D0@D1rt50rt60rrrrrt50r@D0@D1rt55rt57rrt50rt47rt45rt40r
 485:    @D0@D1rt50rt60rrrrrt30r@D0@D1rt50rrrt40rt35rt30rr
 486:    @D0@D1rt50rt60rrrrrt30r@D0@D1rt50rt60rrrrrt50r
 487: /---4.4
 488:    @D0@D1rt55rt65rrrrrt50r@D0@D1t55rt60rt70rr@u42a&a&a&t30a
 489:    @D0@D1t70@u32g&g&g&gf&f&f&f@D0@D1t67e&t65e&t62e&t60et57d&t
55d&t50d&t40d
 490:    @D0@D1t60rrrrt55rt50rt45rt35r@D0@D1t40rt50rt60rrrrrr
 491:    @D0@D1rrrrrrrt40r@D0@D1rt50rt70rrrrrr
 492: /---4.5
 493:    @D0@D1rrrrrrrr@D0@D1t65rrt60rrt55rrt50rr
 494:    @D0@D1rrrrrrrt40r@D0@D1rt50rrrrrrt40r
 495:    @D0@D1t55rrrrrrrt45r@D0@D1rt55rrrrrt45rt35r
 496: /---5.1
 497:    @D0@D1t50rt60rt70rrrrrt40r@D0@D1rt50rt70rrrrrt50rt40r
 498:    @D0@D1t60rt55rt50rt45rrt40rt35rt30r@D0@D1t55rrrrt50rt57rt4
5rt30r
 499:    @D0@D1rt50rrrrrrt45r@D0@D1t47rt50rrrrrrt40r
 500: /---5.2
 501:    @D0@D1rrrrrrrt40r@D0@D1rrt35rrt30rrrt70r
 502:    @D0@D1t40rrt45rrt50rt60rrt50r@D0@D1t55rt60rt65rrrrrt50r
 503:    @D0@D1rt60rrrrrrt55r@D0@D1t60rt65rrr@u74a&a&a&t30a
 504: /---5.3
 505:    @D0@D1t70@u70g&g&g&g@u64f&f&f&f@D0@D1t67@u52e&t65e&t62e&t6
0et55@u44d&t50d&t45d&t40d
 506:    @D0@D1rt60rrrt55rt50rt45rt40r@D0@D1rt60rrrrrrr
 507:    @D0@D1rrrrrrrt40r@D0@D1rt50rt70rrrrrr
 508: /---5.4
 509:    @D0@D1rrrrrrrr@D0@D1t67rt65rt62rt60rt57rt55rt52rt50r
 510:    @D0@D1rrrrrrrt40r@D0@D1rt50rrrrrrt40r
 511:    @D0@D1rt50rrrrrrt40r@D0@D1rt50rrrt45rt40rt35rt30r
 512: /---5.5
 513:    [K.SIGN +c,+d,+f,+g]
 514:    @D0@D1rt220rrrt190rrrr@D0@D1rrrrrrt160rt130r
 515:    @D0@D1rt220rrrt190rrrr@D0@D1rrrrrrt170rt150r
 516: /---6.1
 517:    @D0@D1t190rrrrrrrr@D0@D1rrrrrrrr
 518:    @D0@D1rrrrrrrr@D0@D1rrrrrr@D0rr
 519: /---6.2
 520:    t120rt220rrrt190rrrrrrrrrrt170rt150r
 521:    t190rt220rrrt190rrrrrrrrrrt180rt160r
 522: /---6.3
 523:    t190rrrrrrrrrrrrrrr
 524:    rrrrrrrrrrrrrrr
 525: /---6.4
 526:    @D1t120@u72g&t122g&t125g&t130gt135@u76f&t140f&t145f&t150f@
D0@D1t155@u80e+&t160e+&t165e+&t170e+t175@u84f&t180f&t185f&t190f
 527:    @D0@D1t130rt125rt120rt110rt100rt102rt105rt110r@D0@D1t120rt
130rt140rt150rrrrr
 528: /---6.5
 529:    @D0@D1t160rt162rt165rt167rt170rrrr@D0@D1rrrrrrrr
 530:    @D0@D1rrrrrrt150rt130r@D0@D1t120rt115rt110rt105rt100rt95rt
90rt85r
 531: /---7.1
 532:    @D0@D1t110@u58o5rg&g&gt120rf&f&f@D0@D1t130re+&e+&e+t140rf&
f&f
 533:    @D0@D1t120rt122rt125rt127rt130rrrr@D0@D1t150rt152rt155rt15
7rt160rt162rt165rt167r
 534: /---7.2
 535:    @D0@D1t170rrrrrrrr@D0@D1t175rrrrrrrr
 536:    @D0@D1t190rrrrrrrr@D0@D1t150rt145rt140rt135rt130rt127rt125
rt122r
 537:    @D0@D1t120rt127rt115rt112rt110rt105rt160rr@D0@D1rt157rt155
rt152rt150rt147rt145rt142r
 538: /---7.3
 539:    @D0@D1t140rt135rt130rt125rt120rt117rt115rt112r@D0@D1t110rt
107rt105rt100rt90rt80rt70rt60r
 540:    @D0@D1t140rrrrt137rrrr@D0@D1t135rrrrt130rrrr
 541: /---7.4
 542:    @D0@D1t127rrrr@D0@D1t125rrrr@D0@D1t122rrrrt120rt110rt100rt
90r
 543:    @D0rt220rrrt190rrrrrrrrrrt160rt130r
 544:    t190rt220rrrt190rrrrrrrrrrt170rt150r
 545: /---7.5
 546:    t190rrrrrrrrrrrrrrr
 547:    rrrrrrrrrrrrrrr
 548: /---8.1
 549:    t120rt220rrrt190rrrrrrrrrrrt140r
 550:    t190rrrrrrrrt160rt190rrrrrt150rt120r
 551:    t190rt220rrrt190rrrrrrrrrrr
 552: /---8.2
 553:    rrrrrrrrrrrt170rt165rt160rt155r
```


▶あ，「大魔界村」が「スパII」に変わってる。次はなんだろう。わくわく。そわそわ。
塩瀬　勇人(19)神奈川県

```
   554:   t180rrrrrrrrrrrrrrr
   555: /---8.3
   556:   rrrrrrrrrrrrrrr
   557:   @D1rr@D0rrrrrr@D1rrrr@D0rrrr
   558: /---8.4
   559:   rrrrrrrrrrt170rt150rt130rt110rt90r
   560:   @D1t170rrrrt190rrrrt185rrrrt180rrrr
   561: /---8.5
   562:   t175rrt172rrt170rrt167rrt165rrt162rrt160rt157rt155rt152r
   563:   t150rt147rt145rt142rt140rt137rt135rt132r@D0@D1t130rt127rt1
25rt122r@D0@D1t120rt117rt115rt112r
   564:   @D0@D1t110rt107rt105rt102r@D0@D1t100rt97rt95rt92r@D0@D1t90
rt87rt85rt82r@D0@D1t80rt75rt70rt65r
   565: /---9.1
   566:   @D0@D1t160rrrrrr@u106e&e&@D0@D1e&ee&e&e&ed&d&
   567:   @D0@D1d&drrrrrr@D0@D1rrrrrrrr
   568:   @D0@D1rrrrrrrr@D0@D1rrrrrrrr
   569: /---9.2
   570:   @D0@D1rrrrrrrr@D0@D1rrrrrrrr
   571:   @D0@D1rrrrrrrr@D0@D1rrrrrrrr
   572:   @D0@D1rrrrrrrrrrrrrrrr
   573: /---9.3
   574:   @D0@D1t160rrt157rrt155rrt152rr@D0@D1t150rrt147rrt145rrt142
rr
   575:   @D0@D1t140rrt137rrt135rrt132rrt130rrt127rrt125rrt122rr
   576:   @D0@D1t120rrt117rrt115rrt112rrt110rrrrrrrr
   577: /---9.4
   578:   rrrrrrrrrrrrrrrr
   579:   @D0@D1t105rrrrt102rrrrt100rrrrt97rrrt95r
   580:   @D0@D1t60rt65rt70rt75rt70rt80rt90rt95r@D0@D1t100rt105rt110
rrt90rt85rt80rt75r
   581: /---9.5
   582:   @D0@D1t100rrrrrrrrrrrrrrrr
   583:   @D0@D1rrrrrrrrrrrrrrrr
   584:   @D0@D1rrrrrrrr@D0@D1rrrrrrrr
   585: /---9.6
   586:   @D0@D1rrrrrrrr@D0@D1rt97rt95rt92rt90rt87rt85rt82r
   587:   @D0@D1t80rt77rt75rt72rt70rt67rt65rt62rt60rt57rt55rt52rt50r
t47rt45rt42r
   588:   @D0@D1t80rrrrt77rrrrt75rrrrt72rrrrt70rrt67rrt65rrt62rrt60r
t57rt55rt50rt40rt30rt20rr
   589:   @D0@D1t35rrrrrrrrrrrrrrrr
   590:   @D0@D1t30rrrrrrrrt20rrrrrrrr@D0
   591:
   592: (p)
```

リスト2　幻想即興曲のカウンタ表示

```
1:00006846 00000000   2:00006846 00000000   3:00006846 00000000
```

リスト3　ヤマトタケル


```
  1: / 「ヤマトタケル 魔空戦神異聞」
  2: /        風の少年 [M10]
  3: /
  4: / 作曲・編曲  手塚 理・Vink
  5: /    販売元    KING RECORDS
  6: /
  7: /  採譜・打ち込み  カシュアンヌ
  8:
  9: .comment SC-55:「ヤマトタケル 魔空戦神異聞」より"風の少年[M10]"
 10: .comment   noted, arranged and programmed by Kaschanne
 11:
 12: /=================== Z-MUSIC Reset
 13: (i)(b1)
 14:
 15: /=================== Allocation & Assignment
 16: (m1,1000)(aMIDI1,1)        / melody (ocarina)
 17: (m2,1000)(aMIDI2,2)        / melody (sine wave)
 18: (m3,1000)(aMIDI3,3)        / strings
 19: (m4,1000)(aMIDI4,4)        / bass
 20: (m5,1000)(aMIDI5,5)        / harmony 1 (soprano)
 21: (m6,1000)(aMIDI6,6)        / harmony 2 (alto)
 22: (m7,1000)(aMIDI7,7)        / harmony 3 (tenor)
 23: (m8,1000)(aMIDI8,8)        / harmony 4 (bass)
 24: (m9,1000)(aMIDI10,9)       / bass drum
 25: (m10,1000)(aMIDI10,10)     / gated snare drum
 26: (m11,1000)(aMIDI10,11)     / cymbal
 27: (m12,1000)(aMIDI11,12)     / snare drum
 28:
 29: /=================== MIDI Module Reset
 30: (t1)
 31:  t60
 32:
 33:  @i$41,$10,$42                    /↓ Exclusive Message 送信後は
 34:                              /  少しウェイトを入れた方がいいらしい
 35:  x$40,$00,$7f, 0  r64    / system reset
 36:  x$40,$01,$10, 3,2,2,2,2,2,2,2,2,0,2,0,0,0,0,0     r64
 37:   / voice reserve
 38:   / (MIDI ch. 10,1,2,…,9,11,…,16 の順であることに注意)
 39:  x$40,$01,$30, 3  r64    / reverb macro:Hall1
 40:  x$40,$01,$33, 0,127,64,40         r64
 41:   / reverb level:0, reverb time:127, reverb feedback:64,
 42:   / send level to chorus:40
 43:  x$40,$01,$38, 7  r64    / chorus macro:Short Delay(Feedback)
 44:  x$40,$01,$3a, 127         r64     / chorus level:127
 45:  x$40,$1a,$15, 2  r64    / MIDI11 for drum part 2
 46:  x$40,$01,$00, $59,$61,$6d,$61,$74,$6f,$54,$61,$6b,$65,$72,$75
,$3a,$4d,$31,$30    r64
 47:   / "YamatoTakeru:M10": SC-55の[ALL]を押すと表示される
 48:   / 16文字のASCIIコードでないと `DT1 Data Error!' になる
 49:  @i$41,$10,$45
 50:  x$10,$00,$00, $59,$61,$6d,$61,$74,$6f,$54,$61,$6b,$65,$72,$75
,$3a,$4d,$31,$30    r64
 51:   / same as ;.sc55_print "YamatoTakeru:M10"
 52:
 53: /=================== Etc.
 54: (t1) @i$41,$10,$42 @j1       @g12
 55: (t2) @i$41,$10,$42 @j1 r8^64 @g12
 56: (t3) @i$41,$10,$42 @j1 r8^64 @g12
 57: (t4) @i$41,$10,$42 @j1 r8^64 @g12
 58: (t5) @i$41,$10,$42 @j1 r8^64 @g12
 59: (t6) @i$41,$10,$42 @j1 r8^64 @g12
 60: (t7) @i$41,$10,$42 @j1 r8^64 @g12
 61: (t8) @i$41,$10,$42 @j1 r8^64 @g12
 62: (t9) @i$41,$10,$42     r8^64 @g12
 63: (t10) @i$41,$10,$42     r8^64 @g12
 64: (t11)                   r8^64
 65: (t12)                   r8^64
 66:
 67: /=================== Tempo
 68: (t1)
 69:  t118
 70:
 71: /=================== Melody (ocarina)
 72: (t1)
 73:  @73 v12 @p80 @e50,0 @k3
 74:  / attack time:+3, release time:+10
 75:  @x$b0, $63,1,$62,$63,$6,65, $62,$66,$6,90
 76:  / vibrato (rate:+11, depth:+11, delay:+3)
 77:  @x     $62,8,$6,75, $62,9,$6,75, $62,$a,$6,67
 78:
 79:  r2       / SC-55 の初期化待ち
 80:
 81:  r1 ^1
 82:
 83:  u90
 84:  |:
 85:  L8 o5 q7 g<ccgg4.a b-4.b-af4c
 86:  >g<ccgg4cd ^2..>g
 87:  g<ccgg4.a b-4.b-af4c
 88:  e-4cgf4dc ^2^>g<cd |
 89:
 90:  [k.sign -e,-a,-b]
 91:  e4.(dc)>b4gf gb2g<cd
 92:  e4.(dc)>b4<fb g2^>g<cg
 93:  g4.(fe)d2 f4.(ed)c4.d
 94:  e2^def gdcd>b!<cdg :|
 95:
 96:  [k.sign]
 97:  e-4cgf4dq8c ^1 ^2
 98:
 99: /=================== Melody (choir)
100: (t2)
101:  @53 v12 @p54 @e50,0 @k-1
102:  / release time:+8
103:  @x$b1, $63,1,$62,$66,$6,72
104:  / vibrato (rate:+12, depth:+16, delay:+3)
105:  /@x     $62,8,$6,76, $62,9,$6,80, $62,$a,$6,67
106:
107:  r2        / SC-55 の初期化待ち
108:
109:  r1 ^1
110:
111:  u95
112:  |:
113:  L8 o3 q7 g<ccgg4.a b-4.b-af4c
114:  >g<ccgg4cd ^2..>g
115:  g<ccgg4.a b-4.b-af4c
116:  e-4cgf4dc ^2^>g<cd |
117:
118:  [k.sign -e,-a,-b]
119:  e4.(dc)>b4gf gb2g<cd
120:  e4.(dc)>b4<fb g2^>g<cg
121:  g4.(fe)d2 f4.(ed)c4.d
122:  e2^def gdcd>ga!b!<d :|
123:
124:  [k.sign]
125:  e-4cgf4dq8c ^1 ^2
126:
127: /=================== Strings
128: (t3)
129:  @49 v12 p3 @e50,0 @k2
130:  / attack time:-2
131:  @x$b2, $63,1,$62,$63,$6,62
132:
133:  r2        / SC-55 の初期化待ち
134:
135:  u90
136:  L8 o5 q4 cd>g<cgcd>g <cgcd>g<cdg
137:
138:  |:
139:  |:3 cd>g<cg>g<cg >b-<c>fb-<f>f<cf :|
140:  >a-b-fgb-fb-<c >g<cd>g<cgcd |
141:
142:  [k.sign -e,-a,-b]
143:  @q3 e4.(dc)>b4gf gb2g<cd
144:  e4.(dc)>b4<fb g2^>g<cg
145:  g4.(fe)d2 f4.(ed)c4.d
146:  e2^def gdcd '>b!g''c>a!''d>b!''gd' :|
147:
148:  [k.sign]
149:  q4 >a-b-fgb-fb-q8<c ^1 ^2
150:
151: /=================== Bass
152: (t4)
153:  @40 v12 p3 @e0,40
154:  y$7e,1   / mono mode
155:
156:  r2        / SC-55 の初期化待ち
157:
158:  r1 ^1
159:
160:  u110
```

▶「BASIC修得への道」の最後の「BASICの次はアセンブラを」というアドバイスはまことに的を射た言葉だと思います。　伊沢　納(20)神奈川県

```
161:  |:
162:  L8 o2 q6 |:3 cccccccc >ffffcccc :|
163:  a-a-a-a-b-b-b-b- <ccccccc>b- |
164:
165:  [k.sign -e,-a,-b]
166:  |: aaaabbbb <eeeeeeee :|
167:  >aaaabbbb gggg<cccc>
168:  eeeeeeee gggggggg :|
169:
170:  [k.sign]
171:  a-a-a-a-b-b-b-<q8c ^1 ^2
172:
173: /=================== Harmony 1 (soprano)
174: (t5)
175:  @103 v8 @p79 @e60,0
176:
177:  r2         / SC-55 の初期化待ち
178:
179:  r1 ^1
180:
181:  u64
182:  |:
183:  L2 o5 @q5 e1 ff e1 ff
184:  e1 ff e-d e2.e-4 |
185:
186:  [k.sign -e,-a,-b]
187:  ed e1 ed e.e4
188:  ed de e1 d1 :|
189:
190:  [k.sign]
191:  e-d4.e8 ^1 ^2
192:
193: /=================== Harmony 2 (alto)
194: (t6)
195:  @103 v8 @p19 @e60,0
196:
197:  r2         / SC-55 の初期化待ち
198:
199:  r1 ^1
200:
201:  u64
202:  |:
203:  L2 o4 @q5 e1 ff e1 ff
204:  e1 ff e-f e2.e-4 |
205:
206:  [k.sign -e,-a,-b]
207:  ef e1 ef e.e4
208:  ef fe e1 f1 :|
209:
210:  [k.sign]
211:  e-f4.e8 ^1 ^2
212:
213: /=================== Harmony 3 (tenor)
214: (t7)
215:  @103 v8 @p109 @e60,0
216:
217:  r2         / SC-55 の初期化待ち
218:
219:  r1 ^1
220:
221:  u64
222:  |:
223:  L2 o3 @q5 g1 b-a g1 b-a
224:  g1 b-a a-a- g2.g4 |
225:
226:  [k.sign -e,-a,-b]
227:  aa g1 aa g.g4
228:  aa gg g1 g1 :|
229:
230:  [k.sign]
231:  a-a-4.g8 ^1 ^2
232:
233: /=================== Harmony 4 (bass)
234: (t8)
235:  @103 v8 @p49 @e60,0
236:
237:  r2         / SC-55 の初期化待ち
238:
239:  r1 ^1
240:
241:  u64
242:  |:
243:  L2 o3 @q5 c1 dc c1 dc
244:  c1 dc c>b- <c2.>b-4 |
245:
246:  [k.sign -e,-a,-b]
247:  <c>b b1 <c>b b.<c4
248:  c>b b!<c c1 >b!1 :|
249:
250:  [k.sign]
251:  <c>b-4.<c8 ^1 ^2
252:
253: /=================== Drums
254: (t9)
255:  @17 v12 @e127,0
256:  / dacay time: +4
257:  @x$b9, $63,1,$62,$64,$6,68
258:  / pitch course
259:  @x $63,$18, $62,36, $6, 66        / MONDO kick
260:  @x          $62,38, $6, 68        / Gated SD
261:  @x          $62,52, $6, 59        / Chinese Cymbal
262:  @x          $62,57, $6, 66        / Crash Cymbal 2
263:  / panpot
264:  @x $63,$1c, $62,57, $6,100        / Crash Cymbal 2
265:  / reverb send level
266:  @x $63,$1d, $62,36, $6,  0        / MONDO kick
267:  @x          $62,38, $6,127        / Gated SD
268:  @x          $62,52, $6,127        / Chinese Cymbal
269:  @x          $62,57, $6, 80        / Crash Cymbal 2
270:
271: (t12)
272:  @49 v10 @e64,0
273:  / attack time:+1, decay time:+50
274:  @x$ba, $63,1,$62,$63,$6,65, $62,$64,$6,114
275:  / pitch course
276:  @x $63,$18, $62,38, $6, 62        / Concert SD
277:  @x          $62,40, $6, 61        / Concert SD
278:  / panpot
279:  @x $63,$1c, $62,38, $6, 64        / Concert SD
280:  @x          $62,40, $6, 64        / Concert SD
281:
282: /=== Bass Drum
283: (t9)
284:  @r1
285:
286:  r2         / SC-55 の初期化待ち
287:
288:  u95
289:  L4 o2 cccc cccc
290:
291:  |:
292:  |:8 cccc :| |
293:
294:  |:8 cccc :| :|
295:
296:  cccc8c8 r1 ^2
297:
298: /=== Gated Snare Drum
299: (t10)
300:  @r1
301:
302:  r2         / SC-55 の初期化待ち
303:
304:  r1 ^1
305:
306:  u70
307:  |:
308:  L4 o2 |:8 r2.d :| |
309:
310:  |:8 r2.d :| :|
311:
312:  r2..d8 r1 ^2
313:
314: /=== Cymbal
315: (t11)
316:  @r1
317:
318:  r2         / SC-55 の初期化待ち
319:
320:  r1 ^1
321:
322:  u100
323:  |:
324:  o3 a1 r1^1^1 ^1^1^1^1 |
325:
326:  'e1a' r1^1^1 ^1^1^1^1 :|
327:
328:  'e2..a'a8 r1 ^2
329:
330: /=== Snare Drum
331: (t12)
332:  @r1
333:
334:  r2         / SC-55 の初期化待ち
335:
336:  L8 o2
337:  z75,-2,-25,-4,-4,+35,-5,+5,-5,+5,-10,+5
338:  d{eded}eded{ed}{re}
339:  z75,-2,-25,-4,-4,+35,-5,+5,-20,+10,+5,-10,+10
340:  d{eded}ede{de}{de}{re}
341:
342:  |:
343:  |:
344:  z75,-2,-25,-4,-4,+35,-5,+5,-5,+5,-10,+5
345:  d{dede}dede{de}{rd}
346:  z75,-2,-25,-4,-4,+35,-5,+5,-20,+10,+5,-10,+10
347:  e{dede}ded{ed}{ed}{re}
348:  z75,-2,-25,-4,-4,+35,-5,+5,-5,+5,-10,+5
349:  d{eded}eded{de}{rd}
350:  z75,-2,-25,-4,-4,+35,-5,-10,+10,-5,+10
351:  e{dede}de{rd}{re}{rd}{re}
352:  :| |
353:
354:  |:
355:  z75,-2,-25,-4,-4,+35,-5,+10,-5,-25,-4,-3,+30,-10,-10
356:  d{eded}ed{re}r16{dede}16d{ed}
357:  z75,-2,-25,-4,-4,+35,+5,-5,-25,-4,-3,+30,-10,-10
358:  e{dede}rd{re}r16{dede}16d{ed}
359:  z75,-10,-5,+10,+5,-20,+20,-5,-25,-4,-3,+30,-10,-10
360:  {ed}{ed}r{ed}{re}r16{dede}16d{ed}
361:  z75,-2,-25,-4,-4,+35,-15,+10,+10,-5,-25,-4,-3,+30,-10,-10
362:  e{dede}{de}d{re}r16{dede}16d{ed} :|
363:  :|
364:
365:  z75,-2,-25,-4,-4,+35,-5,+5,-5,+5,-10,+5
366:  d{edde}dede{de}'de' r1 ^2
367:
368: /=================== Performance
369: (p)
```

リスト4 ヤマトタケルのカウンタ表示

```
 1:000015DB 00000000   2:000015DB 00000000   3:000015DB 00000000   4:000015DB 00000000
 5:000015DB 00000000   6:000015DB 00000000   7:000015DB 00000000   8:000015DB 00000000
 9:000015DB 00000000  10:000015DB 00000000  11:000015DB 00000000  12:000015DB 00000000
```


▶まだろくすっぽ読んでないのにマニュアルがなくなったー。全然わからないー(本体付属の5冊)。
安居院　健治(17)東京都

リスト5　きまぐれ オレンジ☆ロード


```
  1: .comment KIMAGURE ORANGE☆ROAD『 ｵﾚﾝｼﾞ・ﾐｽﾃﾘｰ 』by  Yasaka
  2:
  3: (i)
  4: (o136)
  5: (m1,6000)(aMidi07,1)   /ＴＶアニメ  きまぐれオレンジ☆ロード
  6: (m2,6000)(aMidi08,2)   /
  7: (m3,6000)(aMidi03,3)   /  主題歌  『オレンジ・ミステリー』
  8: (m4,6000)(aMidi04,4)   /   作詩    売野雅勇
  9: (m5,6000)(aMidi05,5)   /   作曲    ＮＯＢＯＤＹ
 10: (m6,6000)(aMidi06,6)   /   編曲    鷺巣詩郎
 11: (m7,4000)(aMidi06,7)   /    歌     長島秀幸
 12: (m8,4000)(aMidi01,8)   /
 13: (m9,4000)(aMidi02,9)   / 参考ＣＤ  KIMAGURE ORANGE☆ROAD
 14: (m10,4000)(aMidi10,10) /                 Singing Heart
 15: (m11,4000)(aMidi10,11) /
 16: (m12,4000)(aMidi11,12) / MIDI MODULE Roland SC-55
 17: (m13,4000)(aMidi11,13) /                           (要 ﾊﾞﾝｸ127)
 18: (m14,4000)(aMidi11,14) /
 19: (m15,4000)(aMidi12,15) / TV ｵｰﾌﾟﾆﾝｸﾞﾊﾞｰｼﾞｮﾝ   1994/08/27
 20: (m16,4000)(aMidi09,16) / CD ﾌﾙｺｰﾗｽﾊﾞｰｼﾞｮﾝ            09/25
 21: (m17,4000)(aMidi13,17) /                  (改)       10/08
 22:
 23: .roland_exclusive 16,$42 ={$40,$00,$7F,$00}
 24: .sc55_reverb $10={05,03,00,100,127,10,0}
 25: .sc55_chorus $10={04,03,54,28,$50,03,$13,0}
 26: .sc55_v_reserve $10={0,1,4,4,2,3,1,4,1,0,2,1,1,0,0,0}
 27: .sc55_Print $10="Orange Mystery"
 28:
 29: (t1,2,3,4,5,6,8,9) @I$41,$10,$42 @C11,127,127
 30: (t10,12,13,15,16,17)  @I$41,$10,$42 @C11
 31:
 32: (t1)     @17  @u110 @v105 @G12 @E84,54   @p84  r2@Y1,99,60  r2
 33: (t2)  r2@6   @u110 @v75  @G12 @E34,44   @p44  r2
 34: (t3)  I8@17  @u110 @v120 @G4  @E94,54   @p54  r2@Y1,99,60  r2
 35: (t4)  r2@92  @u110 @v70  @G2  @E44,24   @p48  r4@Y1,102,54 r4
 36: (t5)     @28  @u95  @v110 @G12 @E84,64   @p44  r1
 37: (t6)  r2@91  @u120 @v75  @G12 @E44,24   @p78  r2
 38: (t8)  r2@35  @u120 @v120 @G12 @E10,34   @p64  @Y1,32,19
 39:       r8@Y1,33,14  r8@Y1,99,48  r8@Y1,100,84  r8@Y1,102,54
 40: (t9) I127@63 @u110 @v105 @G12 @E10,24   @p64  @Y1,32,39
 41:       r4@Y1,99,56  r2.
 42: (t10) x$40,$10,$15,$01 @17 @v127 132 r2 @E127,127 r16
 43: /  Volume        Panpot        Rev-depth    Cho-depth    Pitch
 44:   r@Y26,35,110 @Y28,35,64  r@Y29,35,00  @Y30,35,54  @Y24,35,67
 45:   r@Y26,38,110 @Y28,38,64  r@Y29,38,10  @Y30,38,64  @Y24,38,66
 46:   r@Y26,40,110 @Y28,40,64  r@Y29,40,05  @Y30,40,74  @Y24,40,60
 47:   r@Y26,42,110 @Y28,42,54  r@Y29,42,05  @Y30,42,40
 48:   r@Y26,49,110 @Y28,49,78  r@Y29,49,20  @Y30,49,40  @Y24,49,62
 49:   r@Y26,46,110 @Y28,46,64  r@Y29,46,05  @Y30,46,20
 50:   r@Y26,64,110 @Y28,64,64  r@Y29,64,05  @Y30,64,20  @Y24,64,65
 51: (t12) r8 x$40,$1A,$15,$02 @1 @v127 132 r8 @E127,127 r8
 52:   r@Y26,28,120 @Y28,28,64  r@Y29,28,127 @Y30,28,00  @Y24,28,62
 53:   r@Y26,37,105 @Y28,37,64  r@Y29,37,127 @Y30,37,00  @Y24,37,72
 54:   r@Y26,44,65  @Y28,44,14  r@Y29,44,00  @Y30,44,40  @Y24,44,62
 55:   r@Y26,50,112 @Y28,50,24  r@Y29,50,54  @Y30,50,24  @Y24,50,68
 56:   r@Y26,48,112 @Y28,48,24  r@Y29,48,54  @Y30,48,24  @Y24,48,68
 57:   r@Y26,47,112 @Y28,47,94  r@Y29,47,54  @Y30,47,24  @Y24,47,68
 58:   r@Y26,45,112 @Y28,45,104 r@Y29,45,54  @Y30,45,24  @Y24,45,68
 59:   r@Y26,43,112 @Y28,43,104 r@Y29,43,54  @Y30,43,24  @Y24,43,68
 60:   r@Y26,54,60  @Y28,54,114 r@Y29,54,20  @Y30,54,20  @Y24,54,68
 61:   r@Y26,82,70  @Y28,82,114 r@Y29,82,05  @Y30,82,00  @Y24,82,60
 62: (t7,11,13,14) r1
 63: (t15)      @8   @u110 @v50  @E74,0    @p78  r4@Y1,100,50r2.
 64: (t16) I127 @100 @u125 @v127 @E24,64   @p64  r4@Y1,33,74r2.@q19
 65: (t17) I127 @100 @u125 @v77  @E24,44   @p64  r4@Y1,33,74r2.@q18
 66:       @B200 r16
 67:
 68: /B0  [イ]-1
 69: (t1,3)   r*191 @q3 @h72 @m88 @A127,117,107,90,75,60,45,30 18
 70: (t2,5,6,7,8,11,12,14) r1
 71: (t4)      r2 @Y1,99,58r2 o4
 72: (t9)      r1 Z100,115 o2 18
 73: (t10,15) r1                  /↓リバーブタイム変更
 74: (t13)     r2.o1@u110'e<c+'*46 x$40,$01,$34,94r*2
 75: (t16,17) @Y1,32,76r8 @Y1,99,58r8 @Y1,100,104r4 @Y1,102,54r2
 76:
 77: /B1,2  [イ]-2
 78: (t1) 18 |:4o5ua4u-8au<c+4>u-5a<u-3_5d*168 r~5|r4:|r*1
 79:      r32@E74,84 r32@A @p64 r32@66 r32@u110 @v110
 80:      M,1 @S,94 H0,0 S,2 @h24,16 @m32o418@q4@B-1366,0,6a16&a16&
 81: (t2) @q318|:4o4u'a4<c+'u-6'a<d'<u'c+4e'>u-8'a<d'<u'df+'*168 r|
 82:      r4:| r32I0@63@E54,64 @v85r32@p74 @Y1,100,110 r32@Y1,99,60
 83:      M,1 @S,94 H0,0 S,2 @h24,16 @m32 r32 @u110 o5 18 @q4d&
 84: (t3) @L0 |:4 o5Z100,115,92,107 c+&<c+4>d&<d8>e&<e4> Z95,110
 85:      d&<d8> _5f+&<f+*168> r8~5|r4:| r16 @A @m r*37
 86: (t4) @L384'a<d' 'f+b' 'dg' 'e1a'*336 132 r @Y1,99,64
 87:      r @3 @u110@v58 @Y1,102,56 r @E44,0 @p54 r64@Y1,100,29 r64
 88:      @L0 @q3o3@u75a&<@u85d&@u110f+&a&<d*216>>
 89: (t5) @q3 @h36 @m64 Z77,85,55,60,,,75,70,65 o3 14
 90:      'a.<c+',1'a<d',2<d8a8 <dc+>af+d8>
 91:      'a.<c+',1'a<d',2<d8b8 <dc+>bf+d8>
 92:      'a.<c+',1'a<d',2<d8g8 <dc+>bg d8>
 93:      'a.<c+',1'a<d',2<e8a8 <dc+>af+ @m @p94 @u65 d8&
 94: (t6) @h144@M48o5f+1& f+1 @Y1,102,54|:3_20f+2.&~20f+4& f+1:|@M
 95: (t7) |:8r1:|
 96: (t8) @q2o218|:15d:|c+ >u-8|:15b:|a |:16g:| |:15a:|<ud&
 97: (t9) @q2|:7d<d>:|d<c+>> |:7b<b>:|b<a> |:8g<g>:| |:7a<a>:|a<d&
 98: (t10) 14 Z120,90 |:16o1b<|d:|o2@u90d8>@u120b8
 99: (t11) [K.SIGN +c,+f]o3@u8518c>@u60fff16f16
100:       |:15ffr|f16f16:| @u85<c8>
101: (t12) [K.SIGN +g]o2116Z40,50,45,40,50|:r1 r4@Y28,44,14
102:       ggg@Y28,44,84ggg@Y28,44,34gggr8. r1 r1:|
103: (t13) |:7r1:| o3@u100116rrdd8cu-15c>u-7b bu-8aau-8g8r8.
104: (t14) o3@u9014|:4f+u-10f+8uf+u-8f+8uf+ rr8rrr8:|
105: (t15) |:7r1:| r2r4.o4 14 d8&
106: (t16,17) 18 o4 Z110,+5,+5,+5,,-8,-7
107:          r1 r@p4d @p44f+@p84a<@p124d>@p74a @p44f+@p4d
108:          r1 r   d @p44f+@p84b<@p94d> @p124b@p84f+@p74d
109:          r1 r@p4d @p34g @p64b<@p84d> @p94b @p104g@p124d
110:          r1 r@p4c+@p24f+@p34a<@p44c+>@p54a @p64f+@p84c+
111:
112: /B3  [A]-1  きらめく海へ Tシャツのまま
113: (t1) a@B0@q4u+8aq5u-8aau-8@q4a@q2a4u+8q8@m132@B-900a&@B-500a&
114:      @B-200a&@B0a& @m16,32,50,70,80,80,70,60@A127a2r4.@A@m48
115:      @q2u@B-1366a&@B-683a&@B0a16& a418u-8q5aaua@q2@B-700_5a*6&
116:      @B-350a*6&@B0~5a.u+8@B-1366_3q8@ma*5&@B-700a*5&@B0~3a*14&
117:      @m16,32,50,70,80,80,70,60@A127a4.r4@A~8@m48@q4uf+q5aa
118: (t2) d@B0@q4u+8dq5u-8ddu-8@q4d@q2d4u+8q8@md&
119:      @m16,32,50,70,80,80,70,60@A127d2r4.@A@m48@q2uc+&
120:      c+4u-8q5c+c+uc+@q2c+4u+8q8@mc+&
121:      @m16,32,50,70,80,80,70,60@A127c+4.r2@A@mr
122: (t3,6) |:8r1:|
123: (t4) |:@L0                          / B16の :| と対応 (B14に | )
124:      |:Z83,93,118,,118,73,85,110,,110 o3
125:      a&<d&f+&a&<d1>> a&<c+&f+&a&<c+1>> a&<c+&f+&a&<c+1>> |
126: (t5,15) |:d.u+7e2^8 ud2u+8eud c+.u+7d2^8 uc+2u+8dc+ |
127: (t7) |:      |:24r1:| @u90r*191                           / 同上
128: (t8) |:   o2|:                                            / 同上
129:      |:16d:| >u-10|:16f+:|||:16e:| <u|:8c:|> u-10|:7a:|<ud&:|
130: (t9) |:   o2|:d<d>Z100,115|:7d<d>:| >|:8f+<f+>:| |        / 同上
131: (t16,17) |: |:o4 Z110,+5,+5,+5,,-8,-7                     / 同上
132:          r1 r@p124d @p94f+@p74a<@p34d> @p4a@p74f+@p94d
133:          r1 r@p124c+@p94f+@p74a<@p34c+>@p4a@p24f+@p44c+ |
134:
135: /B4  [A]-2  君は翔んだね キスをよけるように
136: (t1) q7u+8@B-1366_6a16&~3@B-683a*5&~3@B0a*31ugu-8f+4uq8@A127
137:      @m16,32,48,64,84,84,44,16g2 r2@A@m32q5ggq6u+8@B-683a*3&
138:      @B0a*21q8u+5@B-683a64&@B0a16.. ruq6gq8g4g16&@B-683_6g16
139:      @B0~6@A127u-8@m16,32,48,64,84,84,44,16e2 r@Ar2_8@m32 u
140:      o4@q2@B-1366,0,8a&
141: (t2) r1 r1 r1 r2r4.u@m32o5@q2d&
142: (t4) Z83,93,118,,118,75,85,110,,110 b&<e&g&b&<e1>>
143:      b&<e&g&b&<e1> c&e&g&<c&e1>> a&<d&e&a&<d4>>
144:      Z75,85,110,,110,83,93,118,,118 a&<c+&e&a&<c+4>>
145:      g&b&<d&g&b4> a&<d&e&a&<d8>> a&<d&f+&a&<d*216>>:|
146: (t5,15) u>b.<d2^8 >b.<d2^8 u+6>g<dg.d8& dc+>b<c+8ud8&:|
147: (t9) |:8e<e>:| <|:4c<c>:|> |:3a<a>:|a<@u115d&:|
148: (t16,17) r1 r@p4d  @p44g  @p74b<@p94d> @p114b @p124g @p94d
149:             r@p124e@p104g<@p84c @p74e  @p54c> @p34g  @p4e
150:             Z105,+5,+5,+5,-7,-8
151:             r@p4e  @p44a< @p74d @p114>b@p84g  @p74er:|
152:
153: /B5  [A]-3  濡れたスカートの 白い花びら
154: (t1) a@B0q5u-8aaua@q2@B-800a*6&@B-350a*6&@B0a.q5au+8q8@B-1366
155:      a*3&@B-683a*3&@B0a16.& @m16,32,50,70,80,80,70,60@A127a2@A
156:      r4.@m32u132@B-1366a&@B-900a&@B-683a&@B0a& a818u-8q6aq5aq6
157:      uaaq8a4132@B-1366a&@B-1000a&@B-650a&@B-400a&
158:      @B0@m32,44,74,94,104,94,84,74@A127a4.r4@A~8@m32q618f+aa
159: (t2) dq5u-8ddud@q2d4q5du+8q8d& @m16,32,50,70,80,80,70,60@A127
160:      d2r4.@A@m32uc+& c+u-8q6c+q5c+uq6c+c+q8c+4c+&
161:      @m32,44,74,94,104,94,84,74@A127c+4.r2@Ar@m
162: (t3) |:                            / B18の :|と対応 (B14に |  )
163:      r1 I0@55r1 @E34,0r1 @Y1,99,64@p74r2@Y1,32,72r*95
164: (t6) r*768
165:
166: /B6  [A]-4  青い水面に 広がってくよ
167: (t1) u+8q8@B-700a*6&@B-300a*6&@B0a.uq6gq8u-8f+4u
168:      @m32,50,70,90,100,90,80,70@A127g2 r4@A@m32q6gu+8q8g.
169:      u@q3f+*30&_8@B-331f+*6@B0q6~8e q8u-8e4q5uf+q8
170:      @m32,50,70,90,100,90,80,70@A127d2r@A <@h36,16@m32@v115r2
171:      q8@B-1366d*4&@B-683d*4&@B0d*28@q2u-8c+.ud
172: (t2) r1 r1 r1 @h36,16@m32@v90r2uq8f+.@q2u-8e.uf+
173: (t3) @u120@v75 M,0 @h72,0 @m16 o3 q8 'b1<d' <'c+1e' 'd1f+'&
174:      @A127,120,110,100,95,90,85,80'd1f+' r*1 @m
175: (t4) Z83,93,118,,118 g&b&<e&g&b1 c+&e&a&<c+&e1>>
176:      a&<d&f+&a&<d*216 @q1 @u100 @v70 o2 18
177:      {fgab<u+5cdeu+5fgu+5ab<u-5cdeu-5fgu-5ab<u-5cu-5d}4.
178:      @E44, @u110 @v60 o4 @q3 'b.<d',0u-8'a.<c+'u'b<d'
179: (t5) u>g.<u+8d2^8 ue.d2^8 u+8f+2.r*11@Y1,99,55 @Y1,32,54 @p74
180:      @E64,74 @v100 r16Z110,,115,115 @L0 o4 @q1
181:      d&f+&a&<d*11> q4d&f+&a&<d*11> @h36 @m16
182:      q8@u110d&@u115f+&@u123a&<d4..r16>@mr4@Y1,99,64r4r*3
183: (t6) r1 r1 I8@29 @Y1,102,64 r2@p54 r*11@E64,54 @v80 o4 r4 @L0
184:      @Y1,99,55r16Z120,,115,115 @q1d&f+&a&<d*11>q3d&f+&a&<d*11>
185:      @Y1,99,60q8 @u110d&f+&@u123a&<d4..r16>r4@A@Y1,100,104r8
186:      @Y1,99,64r8@E34,44r*3
187: (t8) o1|:8e:| |:8a:| <u|:16d:|
188: (t9) o1Z100,115|:4e<e>:| |:4a<a>:| <|:8d<d>:|
189: (t15) |:                              /B20の :|と対応 (B14に |)
190:       @u110o3g.<u+6d2^8 e.d2^8 u+8f+2.r r1
191: (t16,17) o4 Z110,+5,+5,+5,-7,-8 r1
192:          r@p4e @p34g<@p54c+@p74e @p94c+>@p114g @p124e
193:          r@p4d @p44f+@p54a<@p74d>@p124f+@p104a<@p94d>
194:          r@p4f+@p34a<@p54d @p94f+@p74dr4       |:9r1:|
195:
196: /B7  [B]-1  片眼閉じて 僕のものに
197: (t1) u+8c+4>q6ubq8@m64@A127f+2r@A r4@q3r4u+8@m32<@B-1366d*3&
198:      @B-1066d*2&@B-683d*3&@B-300 d*2&@B0d*26@q2uc+.d u+8c+4>
199:      uq6bq8@m64 u-6@A127g2r@A @mr1u
200: (t2) |:u+8@q2e4q6udu-6q8@A127@m64d4.r4@A |
201:      r2@q3u+8@m32f+.@q2ue.u+8@q4f+:| @mr1u
202: (t3) 18r I8@17 r@E54,54 r@p78 r@A127,122,117,112,102,82,62,45
203:      @v105 @q3 @L0 Z125,95,110,120,90
204:      |:@h,16r4.o5c+&e&a&<c+&e*35>r16 @h,12c+&e&a&<c+&e8> |
205:      Z120,80,105,122,105 @h,16d&g&b&<d&g*37>r16r2 r2:|
206:      @A127,122,117,112,102,82,72,60 o5 @h54,12 @m100 @D1
207:      Z127,75,105,122,105 d&g&b&<d&g*37& r2>@D0r16@A @m
208: (t4) @q3@v65u+8'a4<c+''f+8b''d8f+'r8 |:@E,54 r4 @L0
209:      Z100,115,125,92,107,117,90,115,127
210:      @B120~5o4e&a&<c+8..>r32 e&a&<c+*23> | g&b&<d*49 @B0
211:      _5 o4 18 @E,0 @u110'b.<d'u-8'a.<c+'u'b<d'
212:      'a4<c+''g8b''d8g'r8:| @L0 g&b&<d*121 @B0 @E,0r8
213: (t5) r1 r2@Y1,100,104r2 @E44,58r1 @p94r*191
214: (t6) @u75@v90o3@h72@m32'f+2..b<d'r8 r1 @u85'd2..gb'r8 r*191@m
215: (t8) >u-10|:16b:| |:16g:| |:16e:| |:24a:|
216: (t9) >|:8b<b>:| |:8g<g>:|
217: (t15) r4@53r2.@E24,0 r1@p50 r2@Y1,100,104r2 r1 @h,144
218:
219: /B8  [B]-2  なってあげるといったのに・・・
220: (t1) u+8<rq4d4q5ud@q3dddd ddr@B0,-1998,48d4.>@B0@S,140@h48,72b
221:      <q8@m48@B-1366,0,8c+& c+1& @A127@B0,-1366,138c+2.r4@A@m
222: (t2) u+8~5rq4f+4q5uf+@q3f+f+f+f+ f+f+r@B0,-1998,48f+4.@B0d
```

▶今年は受験。もうそろそろ封印をしないと，来年は浪人になってしまうかもしれない。「スパII」を買ってから封印をしようかな。　細江　洋充(18)長野県

```
223:       132r@31 r@Y1,99,64 r@E14,34 @u120 @v55 r*4
224:       o2q6'a<e'*25@q10'a<e''a4<e''a<e'@q2'a<e'*23 @L0
225:       q6'a<e'*25@q10a<e&@u105a4>~5@u120a&<e&@u90a4>
226:       Z120,,95@q6a&<e&a4>@q5a&<e&a*23>
227: (t3) r4 I0 @53 @p48 r4@Y1,100,104 @Y1,99,62 r2 @E24,44
228:       r1 @u120@v95q8o3a1 @v110<c+*191
229: (t4) @3 @v65 q7 o3 @L0 Z90,75,110,,110,82,67,102,,102
230:       b&<d&e&g&b1> b&<d&e&g&b8> b&<d&e&g&b8>r8
231:       @u90b&<@u75d&@u110e&g&b4.>r*46 q6u-5 ~10|:'a1<c+e':|u+10
232: (t5) o4@u110@v75q8@B-4096,0,0'dgb'*4,0&'dgb'*140r4
233:       q7@B-2049,0,0'dgb'*4&'dgb'*20@B0'd8gb'r8q8'dgb'*119
234:       q7|:@u110'c+ea'*25@u85c+&e&@u105a*25r*142 | ~7:|
235: (t6) @E44,58@u115@v90 @Y1,100,64 @p34
236:       o4q8@B-4096,0,0'dgb'*4&'dgb'*140r4_6
237:       q7@B-2049,0,0'dgb'*4&'dgb'*20@B0'd8gb'r8q8'dgb'*119
238:       q7|:@u115'c+ea'*25@u90c+&@u115e&a*25r*142 | ~5:|
239: (t9) |:8e<e>:|Z100,120|:4a<a>:||:4@Y1,32,38a<@Y1,32,58~5a_5>:|
240: (t15) r1 r2@Y1,32,80@Y1,99,60@u120@v95r2 o4q8a1 @v110a*191
241:
242: /B9  [B]-3  O h  B a b y
243: (t1) @B0o4@u115@v115r2rq6ab<c+
244: (t2) Z120,,95@q6a&<e&a*25>a&<e&a4>a&<e&a*49
245:       |:                                  /B23の :|と対応 (B14に |)
246:       r8@63 @E54,64 r16@Y1,99,60 r16@u110 @v85 @p54r8
247: (t3) @h,144@A127,122,117,112,107,102,97,92e1r*1
248: (t4) 18|:'a<c+e'*25:|r'a<c+e' r@Y1,100,44 @v76 r@E24,0 r@p70 r
249: (t5) |:                                   /B22の :|と対応 (B14に |)
250:       @u115'c+ea'*25~10Z100,120,120 c+&e&a*25~5r8q6c+&e&a8
251:       r4@Y1,32,64 r8@Y1,100,64 r16@E54,r16
252: (t6) |:                                                  / 同上
253:       @u115'c+ea'*25~10Z100,120,120 c+&e&a*25~5r8q6c+&e&a8
254:       r8I0@51@u120@v53r8@E54,24r8@Y1,32,74r*23
255: (t9) Z100,125 |:3@Y1,32,38|a<~5@Y1,32,68a_5>:| a<a>a<a
256: (t15) @A127,123,117,112,107,102,97,92a1r*1
257:
258: /B10  [C]-1  T e l l  M e , T e l l  M e  きまぐれだね (☆t1)
259: /              (Tell me love you,tell me that you need me)
260: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4u-8d*23
261:       u@q2@B-1000,0,6_6e*18&~6@m48e*31@m@q3@B0,-683,28d.&
262:       d16@B0q6c+>@q3 @B0,-2049,36a4u-8@B0@m32q8f+& f+4.ur@mr2r1
263: (t2) |:o5@u110@q2'c+4e'q6'd8f+'@q2'c+4e'@q4'd4f+'r8 r1 r8
264:       @L0~8q6Z110,95d&f+4q4d&f+8q2d&f+8d&f+8r8q6Z110,100d&f+4
265:       q4d&f+8q2Z100,95c+&e8>Z100,88b&e8 Z110,80q8d&f+8.@q1
266:       @u90d&@u70f+16q8@u100c+&@u75b&e8@q1>@u90b&<@u70d8_8  *
267: (t3,15) r*768 @A r*768 r*768
268: (t4) @L0 |: Z80,90,100,,100,72,82,92,,92
269:       o3q6a&<d&f+&a&<d4>>q4a&<d&f+&a&<d8>>q6a&<d&f+&a&<d4.>>r4
270:       Z80,90,100,,100,72,82,92,,92
271:       a&<d&f+&a&<d4>>@q3a&<d&f+&a&<d2.>> q6f+&b&<d&f+&b4>
272:       q4f+&b&<d&f+&b8>q6f+&b&<d&f+&b4.>r4 Z80,90,100,,100
273:       f+&b&<d&f+&b4.>q8f+&b&<d&f+&b4.>r4
274: (t5) 18|:r1 r1 @u100@v80o3@q3u-15raba<ud&@B-683d4.
275:       >r@B0u-15aba<ud&@B-683d@B0>b4 |
276: (t6) M,1 @S,200 H0,0 S,2 @h,12o5q8a1 ~8<a2f+4d4 c+1 >b1&
277: (t7) o4@q2a1 <a2f+4d4 c+1 >b1&
278: (t8) |:u o2|:16d:| >u-10|:16b:| | |:16g:| |:16a:|:|
279: (t9) |:Z100,115o2|:8d<d>:| >|:8b<b>:| | |:8g<g>:| |:8a<a>:|:|
280: (t16,17) |: o4 Z110,+5,+5,+5,,-8,-7
281:              r1 r@p4d@p34f+@p54a<@p74d>@p94a@p124f+@p94d
282:              r1 r@p4d@p24f+@p34b<@p54d>@p74b@p94f+ @p124d |
283:
284: /B11  [C]-2  T e l l  M e,  T e l l  M e  夏の天使 (★t1)
285: / (Tell me lover,tell me that you love me) O h  B a b y
286: (t1) <|:@q2@B-1366c+*4&@b-683c+*4&@B0c+*40|q6d:| @q4d4q5u-8d
287:       @h30@m84@q3ue4@m@B0,-683,30d.&d16@B0q5u-8c+q8d.
288:       @B0,-683,32d.&~10d& @m32d2r4._10  u>@B0@mr2q6rab<c+
289: (t2) o5@u110@q2'c+4e'q6'd8f+'@q2'c+4e'@q4'd4f+'r8 r1 r8 |
290:       ~8q6Z110,95e&g4q4e&g8q2e&g8e&g8r8 q6Z110,100e&g4
291:       q4e&g8q2Z110,95d&f+8Z110,88c+&e8q8e&g8d&f+8c+&e8.r16_8:|
292: (t4) Z80,90,100,,100,72,82,92,,92 q6g&b&<d&g&b4>q4g&b&<d&g&b8>
293:       g&b&<d&g&b4.>r4 | Z85,95,105,,105
294:       g&b&<d&f+&a4>@q3g&b&<d&g&b2.>Z80,90,100,,100,72,82,92,,92
295:       q6|:a&<c+&e&a&<c+4.>>a&<c+&e&a&<c+4.>>r4:|:|
296: (t5) r1 r1 ru-15aaa<ue&@B-1366e4.>
297:       u-15@B0raaa<ue&@B-1366e@B0c+4:|
298: (t6) b1 b2<b8g4f+8 e1& @A127e1
299: (t7) b1 b2<b8g4f+8 e1& e1
300: (t16,17) r1 r@p124d@p94g @p74b<@p44d>@p24b  @p4g  @p34d
301:              r@p4c+ @p24f+@p34a<@p44c+>   ar @p124c+
302:              r@p124e@p4a< @p34c+@p54e @p24c+>@p14a @p34e:|
303:
304: /B12  [C]-3  T e l l  M e,T e l l  M e  恋をしてる (◆t1)
305: /              (Tell me love you,tell me that you need me)
306: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| q6d4@q5u-8d
307:       u+10@q2@h18@m64e*40&@me*8u@m@q3@B0,-683,24d*30&d*18 @B0
308:       q5c+>u-8@q2@B0,-1366,38a4@B0@m48q8f+& f+4.r@mr2 r1u
309: (t6) @Ao5_8a1 ~8<a2f+4d4 c+1 >b1&
310: (t7) o4a1 <a2f+4d4 c+1 >b1& b*384 r1 r*193
311:
312: /B13  [C]-4  T e l l  M e,T e l l  M e  君はミステリー (●t1)
313: /              (Tell me lover,tell me that you love me)
314: (t1) <|:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4@q8d @q2
315:       @h24,16@m88e4@m@q3@B0,-683,32d.&d16q4@B0c+@q2c+4u+8 @S,94
316:       q8@B-683d32&@B0d16.& @m32d1& @A127d2r2@A
317: (t2) @L0o5~8q6Z110,95d&f+4q4d&f+8q2d&f+8d&f+8r8q6d&f+4
318:       q4d&f+8q2d&f+8d&f+8q4d&f+8d&f+8d&f+8r8_8
319: (t3) 12r@Y1,100,114r@Y1,99,44 r@p64r@Y1,32,64 rI8@29 r@E34,74
320:       @u120@v127 r r8o3q8@B-4096,0,0d*6&@B0,-4096,36d*42@B0c8
321: (t4) Z90,100,,100,80,90,100,,100 q6a&<c+&e&a4>@q3g&b&<d&g&b2.>
322:       Z80,90,100,,100 q6|:a&<d&f+&a&<d4.>>a&<d&f+&a&<d4.>>|r4:|
323:       r16 @Y1,100,29 r16 @E54,0 18 r
324: (t5) r1 r1 o3u-15a4.<d^2 u+10e4.d2r8
325: (t6) @A127b*384@Ar1 r4I0@49@p54@Y1,32,64r4@E44,44@Y1,99,54r*95
326: (t8) o1|:8g:| |:8a:| <u|:15d:|c
327: (t9) o1Z100,115|:4g<g>:| |:4a<a>:| <|:7d<d>:|o2d<c>
328: (t15) r4@8 r4@u110@v50 r4@E74,0 r4@p78 r4@Y1,100,50r2.
329:       r4@Y1,32,64 r4@Y1,99,63 r2 r1
330: (t16,17) o4 Z110,+5,+5,+5,,-8,-7 r1
331:              r@p14c+@p54g @p74b<@p114c+>@p74a@p44f+@p14c+
332:              r@p54d @p74f+@p94a<@p124d> @p84a@p54f+@p14d
333:              Z105,+5,+5,+5,120 r@p34d@p54f+@p74a<@p94d@p124f+r4
334:
335: /B14  [間奏へ]
336: (t1) @mr1 @Ar1 r1 r1
337: (t2,7,15,16,17) | r*768
338: (t3) | @q2>u-12b-1& b-2^8q8@B-4096,0,0b-*6&b-*42<@B0c8 u+6d1
339:       r4@G12 r4@p58 r4I0@81 @E54,34 r8@Y1,99,52 @Y1,100,104r8
340:       M,1 @S,200 H0,0 S,2 @h36,12 @v85
341: (t4) | o3 @u110@v65 @q3 'b-1<df'*384 <'d4..f+a' @p68 @v60 @q1
342:       o2@u110{defgab<~5cdefgab<~5cdefgab<cdefe}2
343:   Z110,105,102,98,95,91,87,83{dc>bagfed}8.r@Y1,100,29@E54, r2.
344: (t5) | @u110@v60o3q8r@D1u-12b-<u+6duf2@D0u-10c
345:       >r@D1b-<u+6duf2@D0r @D1>u-12du+6f+ua<d2@D0r*23 @h,36
346:       @A127,122,117,112,107,100 @p64 @E,34 o3 @u100 @v80 q8
347:       @B-1366,0,0'a<df+a<d'*6,0&@B0,-8191,24'a<df+a<d'*43,0 r2.
348: (t6) | @v55o3'b-<dfb-<df'*288& @A127'b-2<dfb-<df'
349:       @A<'d2.f+a<df+a' r*2 r16I8 @29 r8@E54,34 r32@Y1,99,64 r*5
350:       o3@u90@v85q8 @B-1366,0,0'a<df+a<d'*6&@B0,-8191,24
351:       'a<df+a<d'*43,0 r4@B0 @p70 r4I0@30 @E,34r8@Y1,99,50
352:       @Y1,100,104r16 @Y1,32,80r16@v100 @h36,12
353: (t8) | u-10>|:15b-:|<uc |:8d:| d.r16r2.
354: (t9) | >|:7b-<b->:|b-<<c> |:4d<d>:| d8.r16r2.
355:
356: /B15  [間奏]-1
357: (t1,2,5,7,15) r*768
358: (t3,6) 18@m32@u110o5q8r4@B-683,0,0{u-15cu+5d}*7@B0ue*17&
359:       @B0,1366,0e&@B1366e4&@B0e>b& b4<c+&@B-1366_25c+16r.~25>
360:       @B-683,0,0q8{u-10au+6b}*6<u@q4@B0c+*21r*9@q6@B-1366,0,0
361:       d*2&d*16e*2&e*16 @B0@q2c+4.>q8@B0,-3415,60a4.@B0e&
362:       @B-1366e&@B-1666e*2&@B-1866e*2& @B-2049e*28&_7@B-1566
363:       e*8&@B-2049e*8>@B0a@B-2049a4&~7@B0a&@B-1366a&@B0a*23
364: (t4) @v50 q5 o3 @L0 Z68,80,115,,115,60,72,107,,107
365:       g&b&<d&g&b8>@p64g&b&<d&g&b4.>@p40g&b&<d&g&b4.>
366:       Z60,72,107,,107 @p54g&b&<d&g&b4>@p40g&b&<d&g&b4>
367:       @p54g&b&<d&g&b8> Z68,80,115,,115@p64g&b&<d&g&b2>
368:       @p54d&f+&b&<d&f+8>@p40 Z60,72,107,,107,68,80,115,,115
369:       d&f+&b&<d&f+4.>@p54d&f+&b&<d&f+4.>Z60,72,107,,107
370:       @p64d&f+&b&<d&f+4>@p54d&f+&b&<d&f+4> @p40d&f+&b&<d&f+8>
371:       Z68,80,115,,115@p54d&f+&b&<d&f+2>
372: (t8) u-10>|:16g:| |:16b:|
373: (t9) Z100,115>|:8g<g>:| |:8b<b>:|
374: (t16,17) @u125 _10 |:7 o4 @p124f+>@p84b<@p34e> @p4b<
375:                          @p34f+> @p84b<@p124b>@p104b:|
376:
377: /B16  [間奏]-2
378: (t1) r*576 @h24,16@m32r2..o4 @u110@v110@q3@B-1366,0,6a16&a16&
379: (t2) r*576 r2@h24,16@m32@p74 @u110@v85 @q3r4.o518 d&
380: (t3,6) uga<q8@B-683,0,0{u-18fgu+8a}*9u@q1@B0b*15<d>@h20@m48b4
381:       u-8q8@h24@m32@B0,-1366,35g*40r*9r@B-683,0,0{u-4fu+6g}*7
382:       u@B0@m48a*65 @m32@B0a&@B-1366a&@B-2049a<@B0d16&@B1300d16&
383:       @B0d*68&@B-683d*2&@B-1366d*2u-6@B0e16&@B1366e16&@B-1366
384:       e4.>u-6@B0b- <uc&@B1366c&@B0c>g4e4&@B-1366_8e&
385: (t4) @p64c+&e&a&<c+&e8> Z60,72,107,,107,68,80,115,,115
386:       @p54c+&e&a&<c+&e4.>@p40c+&e&a&<c+&e4.> @p54d&f+&a&<d&f+4>
387:       Z60,72,107,,107 @p64d&f+&a&<d&f+4> @p54d&f+&a&<d&f+8>
388:       Z68,80,115,,115 @p40d&f+&a&<d&f+2> @p54g&b&<d&g&b8> @p64
389:     Z60,72,107,,107,68,80,115,,115g&b&<d&g&b4.>@p40g&b&<d&g&b2>
390:       @p54@u64b-&<@u80d&@u107f&b-&<d4>>Z73,90,115,,115
391:       b-&<d&f&b-&<d8>c&e&g&<c&e4> @u81c&@u96e&@u121g&<c&e8
392:       r16 @u110 @v58 @Y1,102,56 r32 @E44,0 r32
393:       @q3 o3 @u83a&<@u93d&@u118f+&a&<d*216>>                  :|
394: (t5) @A@Br*576 @p94 @u65@v110 r2@E84,6414r.o4 @q3d8&
395: (t7) r*768                                                     :|
396: (t8) |:8a:| <u|:8d:|> u-10|:8g:| b-b-b-b-<ucccd& :|
397: (t9) |:4a<a>:| <|:4d<d>:| >|:4g<g>:| b-<b->b-<b-c<c>c@u115d&:|
398: (t15) r*576 r2..o414@q2d8&
399: (t16,17) o4u-20@p124f+>u-20@p84b<u-20@p34e>u-20@p4b<u-10@p34f+
400:              >@p84b<@p124b>@p104b~10                          :|
401:
402: /B17  [D]-1  夏が終われば さよならよって
403: (t1) a@B0@q4aq5u-8aq6aau@q2a4q8@m132@B-900a&@B-683a&@B-331a&
404:       @B0a& @m16,32,50,70,80,80,70,60@A127a4.r2@A@m48u@B-1366a&
405:       @B-683a&@B0a16& a818@q4aq5aaq7u+8a@q2uarq8@B-1366132a&
406:       @B-900a&@B-683a&@B-300a& @B0@m16,32,50,70,80,80,70,60
407:       @A127a4.r4@A ~8@m48q618f+q5aa
408: (t2) d@q4dq5u-8dq6ddu@q2d4q8@md8& @m16,32,50,70,80,80,70,60
409:       @A127d4.r2@A@m48uc+& c+@q4c+q5c+c+q7u+8c+@q2uc+rq8c+&
410:       @m16,32,50,70,80,80,70,60@A127c+4.r@Ar2@m
411: (t3,6) @A127e1 r4@A@mr2.@B0 r1 r1
412: (t5,15) d.u+7e2^8 ud2u+8eud c+.u+7d2^8 uc+2u+8dc+u
413:
414: /B18  [D]-2  悪戯っぽく 僕を見つめたね
415: (t1) q7u+8@B-1366_8a*6&@B-1000a*6&~4@B-683a*5&~4@B0a*31uq4gq7
416:       u-8f+4uq8@m16,32,52,72;82,82,72,52 @A127g2 r2@A@m32q5f+g
417:       q6u+8a q8u+6@B-683a64&@B0a16..ruq6gq8f+4g*21&@B-683g*9&
418:       @B-2049@m16,32,52,72,82,92,72,52@A127g4... r4.@A@B0r4_8
419:       @m32o4@q4@B-1366,0,8a&
420: (t2) r1 r1 r1 r2r4.u @m32o5@q4d&
421: (t3) r1 @Y1,32,64r1 @Y1,100,64 12r@Y1,99,64r @G4r@Y1,100,64r:|
422: (t5,15) >g.<u+8d2u+6f+8& f+.u-6d2^8 u>g<dg.u+6d8&duc+>b<c+8d8&
423: (t6) r1 r1 @Y1,32,64r1 @Y1,100,64r2@Y1,99,64r2
424:
425: /B19  [D]-3  渚に倒した バイクに映る
426: (t1) a@B0u-8@q3aq6uaq5a@q3a@q4aq6aq8u+8@B-683a32&@B-300a32&@B0
427:       a16& @m16,32,50,70,80,80,70,60@A127a2r4.@A@m48u@q2@B-1366
428:       a32&@B-683a32&@B0a16& aq6u-8auaa4@q2@B-1366a32&@B-900a32&
429:       @B-683a64&@B-300a64&@B0@q4a^32 q6uaq8@B-683a32&@B-300a32&
430:     @B0a16&@m16,32,50,70,80,80,70,60@A127a4.r@A~8@m48q7uf+q5aa
431: (t2) du-8@q3dq6udq5d@q3ud@q4dq6dq8u+8d&
432:       @m16,32,50,70,80,80,70,60@A127d2r4@A@m48r@q2uc+&
433:       c+q6u-8c+uc+c+4@q4c+4uq6c+
434:       q8c+&@m16,32,50,70,80,80,70,60@A127c+4.r2@A@m
435: (t5,15) d.u+8e2^8 ud2u+8ed uc+.u+8d2u+6e8& e2u-6dc+
436: (t6) r*768
437:
438: /B20  [D]-4  空に落ちてく 君の涙             (t5,6)  B6と同じ
439: (t1) u+8q6@B-683a32&@B-300a32&@B0a.uq5g@q2u-8f+4u+16 q8
440:       @m16,32,50,70,90,90,70,60@A127g*84r@A ur.@m32q6gu+8q7g.
441:       q6uf+.u-8eq7e.q8 @m16,32,50,70,90,90,80,70@A127d2r@Ar. <
442:       @h36,16@m32 ~5r2u@B-1366d*4&@B-683d*4&@B0d*28@q2u-8c+.ud
443: (t2) r1 r1 r1 @h36,16@m32@v90r2q8uf+.@q2u-8e.uf+
444: (t5) u>g.<u+8d2^8 ue.d2^8 u+8f+2.r*11@Y1,99,55 @Y1,32,54 @p74
445:       @E64,74 @v100 r16Z110,,115,115 @L0 o4 @q1
446:       d&f+&a&<d*11> q4d&f+&a&<d*11> @h36 @m16
447:       q8@u110d&@u115f+&@u123a&<d4..r16>@mr4r*3
448: (t6) r1 r1 I8@29 @Y1,102,64 r2@p54 r*11@E64,54 @v80 o4 r4 @L0
449:       @Y1,99,55r16Z120,,115,115 @q4d&f+&a&<d*11>q3d&f+&a&<d*11>
450:       @Y1,99,60q8 @u110d&@u115f+&@u123a&<d4..r16>r4@A@Y1,100,104r8
```


▶あれ、10月号に夫婦で載ってしまったような気がするなぁ……。べつに座布団にしたわけじゃないぞ。 渡辺 祐介(20)富山県

```
451:      @Y1,99,64r8@E34,44r*3
452: (t15) :|
453:
454: /B21  [E]-1  分からないね 君って娘は          (t5,6)  B6と同じ
455: (t1) u+8@q2c+4>uq6bq8@m64@A127f+2r@A r2@m32<@B-1366d*5&@B-683
456:      d*4&@B0d*27@q2u-8c+.u+8@q4d u+8@q2c+4>uq6bq8@m64@A127g2
457:      r@A @mr1
458: (t2) |:u+8@q2e4q6udq8@A127@m64d4.@Ar4 |
459:      r2@m32f+8.@q2u-8e8.u+8@q4f+8:| @mr1
460: (t5) r1 r2@Y1,100,104r2 @E44,58r1 @p94r*191
461: (t6) @u75@v90o3@h72@m32'f+2..b<d'r8 r1 @u85'd2..gb'r8 r*191@m
462:
463: /B22  [E]-2  優しさだけじゃ縛れない
464: (t1) u+10<rq5du-6dd@q3dq5u+6du-6dd d@q3d4q8u+6@B0,-2049,48d4.>
465:      @B0u-5b<@S,140@h48,72@m48u+5c+& c+1&
466:      @A127@B0,-1366,138c+2.r4@A@B0
467: (t2) u+10rq5f+u-6f+f+@q3f+q5u+6f+u-6f+f+ f+@q3f+4q8u+6
468:      @B0,-2049,48f+4.@B0d r32@31r32@Y1,99,64r32@E14,54@v62r*4
469:      @u120o2q6'a<e'*25@q10'a4<e''a4<e'¯5'a4<e'@q2'a<e'*23@L0
470:      ¯10'a<e'*25@q10@u120a&<e&@u105a4>_5Z110,120,90a&<e&a4>
471:      ¯2 Z100,120,95a&<e&a4>¯3@q2Z100,120a&<e&a*22>
472: (t5) @u110@v75o4q8@B-4096,0,0'dgb'*4,0&'dgb'*140r4 @q2
473:      @B-5464,0,0'dgb'*5&'dgb'*19@B0q7'd8gb'r8q8'dgb'*119
474:      q7|:@u110'c+ea'*25@u85c+&e&@u105a*25r*142 | ¯7:|         :|
475: (t6) @E44,58@u115@v90 @Y1,100,64 @p34
476:      o4q8@B-4096,0,0'dgb'*4&'dgb'*140r4 @q2
477:      @B-5464,0,0'dgb'*5&'dgb'*19@B0q7'd8gb'r8q8'dgb'*119
478:      q7|:@u115'c+ea'*25@u90c+&@u115e&a*25r*142 | ¯5:|         :|
479:
480: /B23  [E]-3  O h  B a b y
481: (t1) @mo4@u115@v115r2rq6ab<c+
482: (t2) Z100,120,115a&<e&a*25>@q8a&<e&a4>a&<e&a*50       :|
483:
484: /ここからは、前の行([C]-1,2,3)からの使い回しが多いです
485: /B24  [F]-1  T e l l  M e , T e l l  M e  きまぐれだね (☆t1)
486: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4u-8d*23
487:      u@q2@B-1000,0,6_6e*18&¯6@m48e*31@m@q3@B0,-683,28d.&
488:      d16@B0q6c+>@q3 @B0,-2049,36a4u-8@B0@m32q8f+& f+4.ur@mr2r1
489:
490: /B25  [F]-2  T e l l  M e , T e l l  M e  君が好きさ (◇t1)
491: (t1) <|:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4@q7d
492:      u+8@h24@m88q8e4@mu@q3@B0,-683,30d.&d16@B0q4u+8c+q8@m48
493:      ud4 @md& @B-683@m32d2r2 >@B0@mr2ruq6ab<c+  /O h  B a b y
494:
495: /B26  [F]-3  T e l l  M e , T e l l  M e  近づくほど (▲t1)
496: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| q6d4@q6u-8d
497:      u+10@q2@h24@m88e4u@m@B0,-683,32d*38&d*10@B0q5c+>u-8@q2
498:      @B0,-1366,36a4@B0@m48q8f+& f+4.r@mr2 r1u
499:
500: /B27  [F]-4  T e l l  M e , T e l l  M e  君はミステリー (●t1)
501: (t1) <|:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4@q8d @q2
502:      @h24,16@m88e4@m@q3@B0,-683,32d.&d16q4@B0c+@q2c+4u+8 @S,94
503:      q8@B-683d32&@B0d16.& @m32d1& @A127d2r2@A
504:
505: /B28  [変調]  O h  B a b y
506: (t1) o4r1 @A r2r K1 @u115 q6ab<c+
507: (t2) r1 r1 K1
508: (t3) @q2o2@u105b-2.u-12r8q6b-8 q8@u95o218rb-ru-5a-ru-5gru-5f
509: (t4) o3@u110@v75@q3'b-1<df' 'b-4<df'u-5'a-4<ce-'u-5'g4b-<d'
510:      u-5'f8.a-<c' r32@E24,0 r32@Y1,100,44 K1 @v76
511: (t5) @u110@v75 o3 18 q8r'b-<df'*168,24 r1 K1
512: (t6) @S,200 @h,0 @v60o3'b-1<dfb-<df'& @A127'b-<dfb-<df'*180
513:      r32I0 @51 r*3@E54,24 r64@Y1,32,74 r*1@p44 K1
514: (t7) r1 r*191 K1
515: (t8) o1@u110|:10b-:|a-a-ggff
516: (t9) o1Z100,115|:4b-<b->:| Z100,120b-<@Y1,32,62b->
517:      @Y1,32,38a-<@Y1,32,62a->@Y1,32,38g<@Y1,32,62g>
518:      @Y1,32,38f< @Y1,32,62f  @Y1,32,38
519: (t16,17) r1 r1
520:
521: /B29  [G]-1  T e l l  M e , T e l l  M e  きまぐれだね (☆t1)
522: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4u-8d*23
523:      u@q2@B-1000,0,6_6e*18&¯6@m48e*31@m@q3@B0,-683,28d.&
524:      d16@B0q6c+>@q3 @B0,-2049,36a4u-8@B0@m32q8f+& f+4.ur@mr2r1
525: (t2) |:4o5@u110@q2'c+4e'q6'd8f+'@q2'c+4e'@q4'd4f+'r8 r1 r8
526:      @L0¯8q6Z110,95d&f+4q4d&f+8q2d&f+8d&f+8r8q6Z110,100d&f+4
527:      q4d&f+8q2Z100,95c+&e8Z110,88>b&<d8 Z110,80q8d&f+8.@q1
528:      @u90d&@u70f+16q8@u100c+&@u75e8@q1>@u90b&<@u70d8_8        |
529: (t4) @L0 |:4 Z80,90,100,,100,72,82,92,,92
530:      o3q6a&<d&f+&a&<d4>>q4a&<d&f+&a&<d8>>q6a&<d&f+&a&<d4.>>r4
531:      Z80,90,100,,100,72,82,92,,92
532:      a&<d&f+&a&<d4>>@q3a&<d&f+&a&<d2.>> q6f+&b&<d&f+&b4>
533:      q4f+&b&<d&f+&b8>q6f+&b&<d&f+&b4.>r4 Z80,90,100,,100
534:      f+&b&<d&f+&b4.>q8f+&b&<d&f+&b4.>r4 |
535: (t5) |:4 r1 r1 @u100@v80o318@q3u-15raba<ud&@B-683d4.
536:      >@B0u-15raba<ud&@B-683d@B0>b4 |
537: (t6) @Y1,99,64 |:4 @A@u120@v53o5q8a1 ¯8<a2f+4d4 c+1 >|b1&
538: (t7) |:4 @u90o4@q2a1 <a2f+4d4 c+1 >|b1&
539: (t8) o1 @u110 |:16e-:| <@u120|:16c:|
540: (t9) o1 @q2 Z90,120|:8e-<e->:| Z100,115<|:8c<c>:|
541: (t16,17) |:4 o4 Z110,+5,+5,+5,,-8,-7
542:           r1 r@p124e-@p94g @p74b-<@p54e->@p34 b-@p4g  @p44e-
543:           r1 r@p4e-  @p44g<@p74c  @p94e- @p124c>@p104g@p84e- |
544:
545: /B30  [G]-2  T e l l  M e ,  T e l l  M e  夏の天使 (★t1)
546: (t1) <|:@q2@B-1366c+*4&@b-683c+*4&@B0c+*40|q6d:| @q4d4q5u-8d
547:      @h30@m84@q3ue4@m@B0,-683,30d.&d16@B0q5u-8c+q8d.
548:      @B0,-683,32d.&¯10d& @m32d2r4._10  u>@B0@mr2q6rab<c+
549: (t2) o5@u110@q2'c+4e'q6'd8f+'@q2'c+4e'@q4'd4f+'r8 r1 r8
550:      ¯8q6Z110,95e&g4q4e&g8q2e&g8e&g8r8 q6Z110,100e&g4
551:      q4e&g8q2Z110,95d&f+8Z110,88c+&e8q8e&g8d&f+8c+&e8.r16_8 :|
552: (t4) Z80,90,100,,100,72,82,92,,92 q6g&b&<d&g&b4>q4g&b&<d&g&b8>
553:      g&b&<d&g&b4.>r4 Z85,95,105,,105
554:      g&b&<d&f+&a4>@q3g&b&<d&g&b2.>Z80,90,100,,100,72,82,92,,92
555:      q6|:a&<c+&e&a&<c+4.>>a&<c+&e&a&<c+4.>>r4:| :|
556: (t5) r1 r1 u-15raaa<ue&@B-1366e4.>
557:      u-15@B0raaa<ue&@B-1366e@B0c+4 :|
558: (t6) b1 b2<b8g4f+8 e1& @A127e1 :|o5b1
559: (t7) b1 b2<b8g4f+8 e1& e1 :|o4b1
560: (t8) |:3o1 @u110|:16a-:| |:16b-:| <@u120|:16e-:| |:16c:| :|
561: (t9) |:3o1|:8a-<a->:| |:8b-<b->:| <|:8e-<e->:| |:8c<c>:|:|
562: (t16,17) r1 r@p124e-@p94a-@p74<c@p54e-@p34c>@p4a-@p34e-
563:           Z110,+5,+5,+5,120,120
564:           r@p124d@p94g  @p74b-<@p54d>@p34b- r @p4d
565:           r@p4f  @p44b-<@p74d  @p94f @p114d>  @p127b-@p94f :|
566:
567: /B31  [G]-3  T e l l  M e , T e l l  M e  恋をしてる (◆t1)
568: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| q6d4@q5u-8d
569:      u+10@q2@h18@m64e*40&@me*8u@m@q3@B0,-683,24d*30&d*18 @B0
570:      q5c+>u-8@q2@B0,-1366,38a4@B0@m48q8f+& f+4.r@mr2 r1u
571:
572: /B32  [G]-4  T e l l  M e , T e l l  M e  君はミステリー
573: (t1) <|:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4@q8d
574:      @q2@h24@m88e4@m@q3@B0,-683,32d.&d16q4@B0c+@q2d4q8d
575:      ;$CC,$06 u@m32c+2^8r4. >r2@mrq6ab<c+
576: /     ↑(全トラックフェードアウト)
577:
578: /B33  [H]-1  T e l l  M e , T e l l  M e  きまぐれだね (☆t1)
579: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| @q4d4u-8d*23
580:      u@q2@B-1000,0,6_6e*18&¯6@m48e*31@m@q3@B0,-683,28d.&
581:      d16@B0q6c+>@q3 @B0,-2049,36a4u-8@B0@m32q8f+& f+4.ur@mr2r1
582:
583: /B34  [H]-2  T e l l  M e , T e l l  M e  君が好きさ (◇t1)
584: (t1) <|:@q2@B-1366c+*4&@b-683c+*4&@B0c+*40|q6d:| @q4d4@q7d
585:      u+8@h24@m88q8e4@mu@q3@B0,-683,22d*26&d*22@B0q4u+8c+q8@m48
586:      ud4 @md& @B-683@m32d2r2 >@B0@mr2ruq6ab<c+  /O h  B a b y
587:
588: /B35  [H]-3  T e l l  M e , T e l l  M e  近づくほど (▲t1)
589: (t1) |:@q2@B-1366c+*4&@B-683c+*4&@B0c+*40|q6d:| q6d4@q6u-8d
590:      u+10@q2@h24@m88e4u@m@B0,-683,32d*38&d*10@B0q5c+>u-8@q2
591:      @B0,-1366,36a4@B0@m48q8f+& f+4.r@mr2 r1u
592: /------------------------リズムパート------------------------
593: (t10) |:                                         /B. D&S. D
594:       |:Z120,90r8b8<d>b<d | |:7o1b<d>b<|d:| o2@u90d8>@u120b8:|
595:       Z120,90|:11o1b<d>|b<d:| o1@u120b8<Z80,85,78,84{dddddd}4.
596:       Z120,90|:9o1b<d:| o118b<@u90@Y24,38,65d@Y24,38,66dd
597:       14|:Z120,90|:8o1b<|d:| o2@Y24,38,65@u90d | Z120,90
598:       @Y24,38,66 |:7o1b<d:| o1b8<@u90@Y24,38,65d8@Y24,38,66
599:       d8d8:| @Y24,38,66 Z120,90 |:8o1b<d:| | |:6o1b<|d:|
600:       o2@u90d8Z85,90d16d16 @u80{du-15du-15du-20du-15du-10d}1
601:       Z120,90|:16o1b<|d:| o2@u90d8@u120>b8 :| |:3o1b<d>:|
602:       o1b8<@u90@Y24,38,65d8 @Y24,38,66d8d8 |:4Z120,90|:8o1b<|d
603:       :| o2@Y24,38,65@u90d @Y24,38,66 | Z120,90|:7o1b<d:|
604:       o1b8<@u90@Y24,38,65d8@Y24,38,66d8d8:|
605:
606: (t11) |:|:@u60|:14ffrf16f16:| | ffff16f16fff<@u85c>:| /ｼﾝﾊﾞﾙ系
607:       |:ffff16f16:| @u85<c>@u60fff16f16 |:6ffrf16f16:|
608:       ffff16f16 |:4ffrf16f16:| @Y28,49,44
609:       |:3<u-25c>ufff16f16ff|rf16f16:|r16 @Y28,49,84r16f16f16
610:       |:@u85<c>@u60fff16f16 |:6ffrf16f16:| ffff16f16
611:       | |:7ffrf16|f16:| r16rfff16f16:| |:4ffrf16f16:|
612:       @Y28,49,44<u-30c>ufrf16f16 @Y28,49,84 |:3ffrf16f16:| |
613:       @u85<c>@u60fff16f16 |:3ffrf16f16:|@Y28,49,44
614:       u-25<c>ufrf16f16ffrf16f16 Z80,90b-8f8r4 @Y28,49,84r2@u85
615:       <c>@u60fff16f16|:6ffrf16f16:| ffff16f16 |:6ffrf16f16:|
616:       @Y28,49,44|:<u-30c>uf|rf16f16:| @Y28,49,84r@u85<c8> :|
617:       |:@u85<c>@u60frf16f16ffrf16f16:|
618:       |:4@u85<c>@u60fff16f16|:6ffrf16f16:|ffff16f16
619:       | |:7ffrf16|f16:| r16rfff16f16:|
620:
621: (t12) |:|:4r1 r4@Y28,44,14ggg@Y28,44,84ggg@Y28,44,34 / PAN Hat
622:       gggr8. r1 r1:| |:9r1:| |:4r1 r4@Y28,44,14ggg@Y28,44,84
623:       ggg@Y28,44,34 gggr8. r1 r1:| | |:r4@Y28,44,14ggg
624:       @Y28,44,84ggg@Y28,44,34gggr8. r1:| |:r1 r4@Y28,44,14ggg
625:       @Y28,44,84ggg@Y28,44,34gggr8.r1 r1:|:| r1 r1 |:7 r1 r4
626:       @Y28,44,14ggg@Y28,44,84ggg@Y28,44,34gggr8. r1 r1:|
627:
628: (t13) |:|:6r1:| @Y28,48,94r1 @u10018<rdrc>rbr4@Y28,48,24 /タム
629:       |:7r1:| @u10018<rd>u-10bu-8au-8gr4. |:5r1:|r2@Y28,47,114
630:       r2 @Y28,48,84r1 @u105o318rdrdrcrc>ru-8bru-8br4@Y28,48,24
631:       r4 |:7r1:| 116r8o3@u100ddcc>u-8bu-8agr8.r4 |:7r1:|
632:       r2@u80o3d4d4 | |:3r1:| r*138x$40,$01,$34,127 o1@u110r*6
633:       'e<c+'*42 x$40,$01,$34,94 r*6q4 |:7r1:|
634:       o3@u100116rrdd8cu-10c>u-10b bu-8aau-8g8r8.u :|
635:       o3@Y28,48,74@u100o318rdrdrcrc >rbrbr4@Y28,48,24r4|:7r1:|
636:       116r8o3@u100ddcc>u-8bu-8agr8.r4 |:3 |:4r1:| | r1r1r1
637:       @Y28,48,74r8o3@u100d4c4>b4r8:|
638:
639: (t14) [K.SIGN -b]o518|:                                    /Shaker
640:       @u90|:16r1:| |:bbr4bbr4 bbr2.:| r*768 @Y24,82,64rbrbr4
641:       @Y24,82,60r4 |:3bbr4bbr4 | bbr4bbrb:| |:bbr4bbrb16b16:|
642:       |:bbr4bbr|b:|r |:bbr4bbrb | bbr4bbrb16b16:| |:bbr4bbr4
643:       bbr4bbrb:| | |:12r1:| :| r1r1 |:4|:3bbr4:|bbrb |:bbr4bbr
644:       |b16b16:|b | |:bbr4bbrb16b16 bbr4bbrb:| :|
645: (p) /演奏
```

リスト6　きまぐれ オレンジ☆ロードのカウンタ表示

```
 1:00006480 00000000   2:00006480 00000000   3:00004F80 00000000   4:00006480 00000000
 5:00006480 00000000   6:0000647F 00000000   7:0000647F 00000000   8:00006480 00000000
 9:00006480 00000000  10:00006480 00000000  11:00006480 00000000  12:00006480 00000000
13:00006480 00000000  14:00006480 00000000  15:00004E00 00000000  16:00006480 00000000
17:0000648C 00000000
```

▶最近薄くなりましたね(なにが?)。　　　　国竹　泰夫(20)東京都

SIDE B

サスペンションをさらに考える

Yokouchi Takeshi 横内 威至

引き続きサスペンションを考える
前回は，サスペンションの基本的な動作の解説を行ったが
今回はさらにサスペンションに含まれている要素を追求していく

ちょっと前にカートを乗りに行った。カートに乗ったとたん「走る車」というのが理解できた。ステアに対するリニアな反応，コーナリングなんかはそこらの一般車とはまったく別物。オーバーめのセッティングなので，ドリフトアウトよりもパワースライドによるスピンが多発してショックであった。チキンな性分が災いして限界近い走行はあまりできなかったのが残念だ。カウル壊すだけで何万もかかるので，コースを攻める前に理性が働く。

そして皆が見ている前だからあまりカッコ悪い真似はできない。「女に下手だといわれたくないものは2つある」と，あるレーサーがいっていたらしいが，俺もそう思う。運転技術は男のプライドだ。誰も見ていないところで練習しよう。常に余裕をもって上手そうに見えるように。バトルで負けても，「俺は走り屋じゃない。流し屋だ」「人を乗せてるときは安全のほうが大事だ」でやりすごす。でも，本当はうまくなりたいぜ。

とまあ，余談はこれくらいにして，今回も引き続きサスペンション関係を考える。走り屋諸君ならば普通に知っている知識だが，理論的な側面まで理解している人はどれくらいいるのだろう。私はまったく理解などしていなかった。というより，走り屋ではないからセッティングによる効果自体をそんなに知らない。本来ならば自分で試してみてから理解する必要があるかもしれないのだが。さらに，ここにきて，学問は実験も重要だということを思い知る。しかしいままでの数々の演習，物理実験なんかはどれも覚えていない。生活にまったく絡まない内容だし，しかたないね。それに金も根性も場所もないから自前では実験不可能。まずは材料となる安い車，そして走って面白い車が欲しいものだ。

それでは，シミュレーションに関係するかどうかはわからないが，サスペンションの細かい影響なんかをしっかり調べてみよう。どこまでを再現できるかはわからない。だが，考えられるかぎりの効果が再現できればベストなのである。

キャンバー

図1のように，路面に対してのタイヤの垂直方向の傾きをキャンバー角という。キャンバーにより，車の走行がどのように変化するかは，バイクの挙動を思い浮かべてもらいたい。タイヤの傾いている方向に向けて力が発生するのである。これをキャンバースラストと呼ぶ。バイクのコーナリングではこの力がサイドフォース（タイヤの発生するコーナリングフォース，キャンバースラストなどタイヤに発生する力の合力）の大部分を占める。これが4輪の自動車になるとまた別で，タイヤと進行方向の角度によって発生するコーナリングフォースが大部分を占めることになる。とはいってもキャンバースラストは無視できるようなレベルではない。サスペンションの構造によっていくらでも変化するため，車の種類によってもその特性が変わってくるからだ。

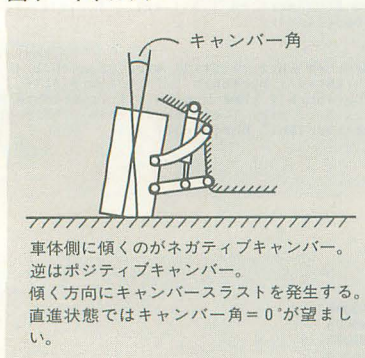
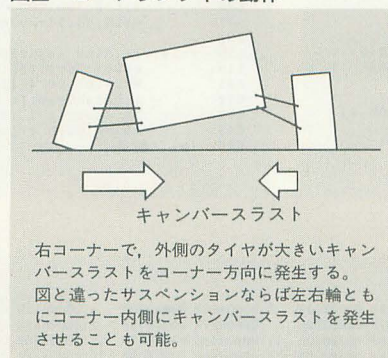
そして，タイヤが車体の外側に傾くのがポジティブキャンバー，その逆がネガティブキャンバーである。これらがどのように影響するかはサスペンションの動きと連動させて考えなければならない。

とりあえず，図1とコーナリング中の図2を見て

図1　キャンバー

車体側に傾くのがネガティブキャンバー。
逆はポジティブキャンバー。
傾く方向にキャンバースラストを発生する。
直進状態ではキャンバー角＝0°が望ましい。

図2　コーナリング中の動作

右コーナーで，外側のタイヤが大きいキャンバースラストをコーナー方向に発生する。
図と違ったサスペンションならば左右輪ともにコーナー内側にキャンバースラストを発生させることも可能。

ほしい。強引なネガティブキャンバー，走り屋の間では「鬼キャン」として親しまれているキャンバー設定である。図１のような例は前輪のサイドフォースが強烈で，ドリフト仕様の定番である。当然，後輪にはボロいタイヤを履かせて簡単にブレークするようにするのが一般的だ。なぜサイドフォースが強烈になるかというと，コーナー内側に向かって，大きいキャンバースラストが発生するからである（これは勝手な解釈なので間違っていたら申し訳ない)。

トーコントロール

図３のように，直進時の車体方向とタイヤの向きがトー角である。必ずしもタイヤが真っすぐ向いているわけではない。これがどれくらい走行に関わってくるかはいまいち理解しにくい。本で調べているだけなので，重要度は不明である。ただし，実際これによってステアリング特性，オーバーやアンダーを調整しているらしいので，サスペンションの動作のひとつとしてしっかりと考えなければならないことは確かだ。

車体前方を内側に向けておくのがトーイン，逆をトーアウトという。実際の設定では大きくても0.2°程度のものらしい。これがどのように走行性能に関わってくるかが図４である。これはコーナリングフォースによってサスペンションの柔軟性がトーコントロールを行う場合の例である。同じような動作は4WSによっても可能だが，サスペンションの特性としてのトーコントロールの話である。とりあえず，結果的にどうなったであろうか。図４では安定性に関わる例であるが，この場合では前輪のサイドフォースよりも後輪のサイドフォースが大きくなっている。後輪にワイドタイヤなどの，より大きいコーナリングフォースが発生する高性能タイヤを履いたのと同じ結果となる。

サスペンションジオメトリー

先月から安易にタイヤと接地について考えてきたが，今月のキャンバー，トーを含めるとサスペンションはより濃厚な内容のものとなってしまう。そして，こうしなければあらゆる車，本来のハンドリングの味わいというのがまったくなくなってしまうことが予想される。

考えてみよう。車のコーナリングはどのように決定されるのだろうか。平坦な路面が一般的なので，平面上でのコーナリングを考える。サスペンションが働くのは，車体がロールしているとき，つまりタイヤの傾きを決定するときである。このときキャンバースラストを考えなければ，ロールによって変化するのは左右の荷重移動だけとなる。トー変化がなければ左右輪のスリップアングル（走行方向とタイヤのなす角）は一緒と考えられるので，前輪，後輪の発生するサイドフォースは左右輪の平均値の２倍となる。

これがどういうことかというと，左右で違ったタイヤを履かないかぎりは，ロールに関係なくスリップアングルだけがコーナリングの挙動を決定することになってしまうのである。結果的に，サスペンションはコーナリングに関係しないことになってしまう。要するにネガティブキャンバーはドリフトに関係なくなってしまうのだ。極端な話，サニーのトレッド，重量配分そして重量をGT-Rと同じにすれば，サニーはGT-Rになってしまうのだ。さらにエンジンと空力なんかを考えればすぐさまサニーがF1に出てしまうかもしれない。

しかし，サスペンションは深みにはまると奥が深すぎる。サスペンションの最大の役割は確実に最良の接地を行い，そして車両を安定させて走らせることにある，と理解している。コーナリングでは車体にロールが起こるのは物理的に当然のことである。加速，減速時にはピッチングするのも当然（厳密にはそのように力が働く，というのが正しい)。

では，車体が定常状態から崩れたとき，どうやってタイヤに最大限の性能を発揮させるか，それがサスペンションジオメトリーなのである。車の設計の基本であり，この特性ゆえにハンドリングの味が出てくるのである。サニーならばファミリーカーになり，GT-Rは極上のスポーツカーになるのだ。FFのCR-Xだってコーナリングするのはサスペンションとほかのバランスを絶妙に生かしているのである。

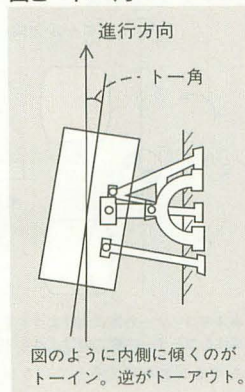
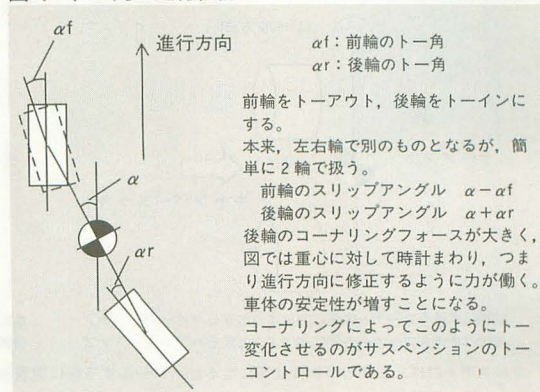
では，サスペンションジオメトリーとは具体的になんなのだろうか。参考文献１によると「幾何学的配置」とされている。要するにスプリングやアーム，リンクなどを含めた，車体とタイヤをつなぐ可動部分の構造のことである。いままで解説してきた図のように，車体がロールすることによってキャンバー

図3　トー角

図のように内側に傾くのがトーイン。逆がトーアウト。

図4　トー角と走行性能

αf：前輪のトー角
αr：後輪のトー角

前輪をトーアウト，後輪をトーインにする。
本来，左右輪で別のものとなるが，簡単に２輪で扱う。
　前輪のスリップアングル　α－αf
　後輪のスリップアングル　α＋αr
後輪のコーナリングフォースが大きく，図では重心に対して時計まわり，つまり進行方向に修正するように力が働く。車体の安定性が増すことになる。
コーナリングによってこのようにトー変化させるのがサスペンションのトーコントロールである。

ハードコア3Dエクスタシー(第14回)

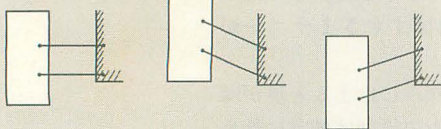
角に変化が生じることは理解できた。

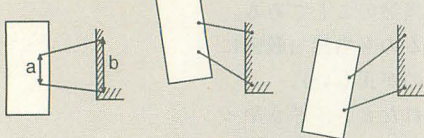
では，いろいろな幾何学的配置によってどのような変化が起こるだろう。図5にいろいろな種類の方式を挙げる。一応，図5はキャンバー角の変化と幾何学的配置の関係となっている。同様に，トー角も幾何学的配置によって変化するように作られている。トー角の変化は前述のトーコントロールとともに，ロールによってサスペンションが動くときにも起こる。この変化によってコーナリングにどのような影響が出るだろうか。キャンバーのところでも出てきたが，また別の側面を考えたのが図6である。ステアリング特性に違いが出てくるのが理解できると思う。

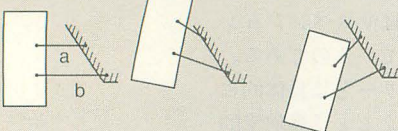
しかし，実際の構造はこんな単純なものではない。いろいろな車のカタログを見ればわかるように，似たような方式ではあっても特性はまったく違っていたりもするし，これに関しては本当にそれぞれの車によって異なっているものである。各車によって重心も違うし，狙いも違うし，またスペースの都合もあれこれある。サスペンションが車によって多種多様なのはしかたがないことなのだろう。

これをシミュレートするというのは嫌なものである。シミュレートするならばロールとキャンバー，トーそれぞれの角度の関係をあらかじめテーブル化させればよいのだが，実際に調べるのは自力では不可能と思ってかまわない。そしてセッティングによっても変化させなければならないから，かなり厄介な部分であろう。困ったものである。できれば省きたいのだが，本来ならば車のクセを最も感じさせる部分だろうから致命的なことにもなるのであろう。もうしばらくは考えてみなければならない。

図5 サスペンションジオメトリー

(1) 等長，平行アーム

キャンバーの変化はまったくない

(2) 等長，非平行アーム

a＜bで図のようにタイヤが上がる状態でポジティブ，
車体が上がる状態でネガティブとなる。
a＞bならば逆の特性になる。

(3) 不等長，平行アーム

a＜bならば中間からネガティブのみ。
a＞bならば中間からポジティブとなる。

4輪接地モデル

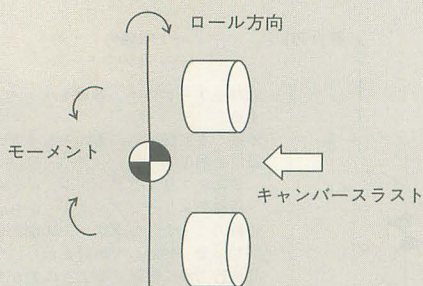
やっとここに到達であるが，いままでの要素をすべて詰め込もうとするとかなり難しい。というより，下手にあらゆる要素を詰めようとするといろいろな矛盾が生じ，不可解な挙動を起こしかねない。簡単に考えられる例は，駄目なピンボールシミュレーションなんかでよく見かける（最近ではピンボール自体見かけないか）玉ブレ。フリッパーで玉を止めておいても，いつまでたっても揺れているような現象と同じことがドライビングシミュレータでも起こりかねないのだ。

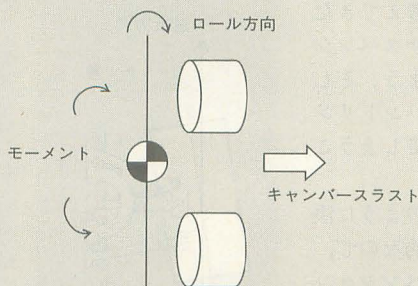
さらに難しさは時分割による部分も大きい。1ターンの時間が決まっていればいろいろな手抜きが可能である。とはいえ，1ターンに1秒だとかの低速度では意味がない。十分に滑らかに感じるスピードならばなんとかツジツマ合わせができる。ちょっとした段差を高速で越えて車が空中に浮き，そのあとの着地によるサスペンションの浮き沈みは接地瞬間からの時間によって変化する。処理が十分に速ければ接地したターンからの減衰振動で問題ないが，1ターンの時間が長ければそれは許されない。とりあえず，レスポンスの問題もあって，それなりに滑らかな動きにしなければならないのは明白なので割り切ってしまってもかまわないのかもしれないが，バランスをとって調節すべきであろう。

またもやこの部分で終わりだが，サスペンションの重要な部分が終わったので来月こそしっかりと考えたいと思う。限界を感じるような走りもしてみないと車の本当の性能はわからない。冬になる前にもう少し熱くなるべきかな。それではまた。

図6 コーナリング中の影響

右輪についてのみを考える。左コーナリング中のキャンバー変化によるステア特性への影響は下のようになる。

(1) ネガティブキャンバー

前輪は車体をコーナー内側に向けようとする＝オーバーステア
後輪は車体をコーナー外側に向けようとする＝アンダーステア

(2) ポジティブキャンバー

前輪は車体をコーナー外側に向けようとする＝アンダーステア
後輪は車体をコーナー内側に向けようとする＝オーバーステア

全体を考えれば，左側の車輪では逆になるが，ロールする側に荷重移動をすることを考えると，バランスは上記のようになる。

〈参考文献〉
1）車両運動性能とシャシーメカニズム，宇野高明著，グランプリ出版
2）自動車の運動と制御，安部正人著，山海堂
3）スポーツカー　理論と設計，コーリン・キャンベル著，二玄社
4）CR-X delsol　カタログ，本田技研，ホンダベルノ

THE SENTINEL

〈対応機種一覧〉●MZ-80 K/C/700/1500 ●MZ-80 B/2000 ●MZ-2500/2861 ●X1 ●X1 turbo/Z ●PC-8001/8801/88 ●SMC-777/C ●PASOPIA/5 ●PASOPIA/7 ●FM-7/77/AV ●MSX/2/2+/turbo R ●PC-286/386/486/9801/98/9821 ●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第152部　シューティングゲーム作成講座(4)

●MOOK化計画次なるステップへ(3)

ここで，S-OS"SWORD"MOOKにおける，現在収録予定のアプリケーションリストを掲載します。

以前にもお話ししたとおり，フリーソフト化計画で協力していただいたものを中心としているため，結構新しいものが多くなっています。

そして，収録予定リストを見ていただければおわかりのとおり，収録予定のアプリケーションの中にはコピーフリーの許可を得ていないものもあります。本来ならば，きちんと許可の取れているものだけを自由に配布したいところです。しかし，やはりS-OSを活用するためには欠かせないアプリケーションがあります。

もしも，コピーフリーという言葉にためらっている人がいましたら，MOOKに収録するだけでも許可していただけませんでしょうか。

また，このリストは担当の独断で作成してあります。このほかにも収録してもらいたいものがあれば，アンケートハガキでお知らせください。

なお，引き続きフリーソフト化計画は募集していますので，自分の作ったアプリケーションをコピーフリーにしていいという人がいましたら，こちらもアンケートハガキで連絡してください。

それでは，連絡をお待ちしています。

●収録予定リスト

以下のリストはジャンルごとに分けて，フリー許可　掲載部：アプリケーション名のように構成してあります。

・マシン語

×61部：デバッキングツールTRADE
×74部：ソースジェネレータSOURCERY
○77部：高速エディタアセンブラREDA
○90部：超多機能アセンブラOHM-Z80
○93部：リロケータブルフォーマットの取り決め
○96部：リロケータブルアセンブラWZD
○97部：リンカWLK
○99部：ライブラリアンWLB

・インタプリタ言語

○28部：FuzzyBASIC
×85部：小型インタプリタ言語TTI
×92部：インタプリタ言語STACK

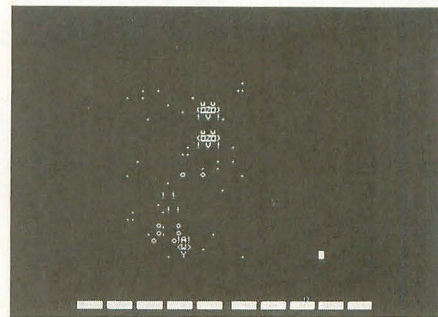
・コンパイラ言語

○44部：FuzzyBASICコンパイラ
○60部：構造型コンパイラ言語SLANG
×81部：超小型コンパイラTTC
×89部：超小型コンパイラTTC++
○106部：実数型コンパイラREAL

・エディタ

○69部：超小型エディタTED-750

・ゲーム

×40部：INVADER GAME
○82部：TTC用パズルゲームTICBAN
○86部：TTI用パズルゲームPUSH BON!

○88部：SLANG用ゲームWORM KUN
○94部：STACK用ゲームSQUASH
○103部：ダイスゲームKISMET
○104部：アクションゲームMUD BALLIN'
○113部：MORTAL
×115部：LINER
○116部：シミュレーションゲームPOLANYI
×117部：カードゲームKLONDIKE
○125部：SLENDER HUL
○129部：BLACK JACK
○133部：REVERSI
○151部：B-GALETS2

●開発ツール

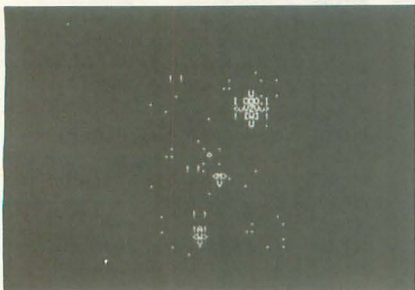
○35部：MACINTO-C
×37部：テキアペ作成ツールCONTEX
○48部：漢字出力パッケージJACK WRITE
○78部：Z80浮動小数点演算パッケージSOROBAN
○119部：COMMAND.OBJ
×123部：グラフィックライブラリGRAPH.LIB
×124部：O-EDIT&MODCNV
○145部：YGCS ver.0.30

1994■インデックス

全機種共通
S-OS"SWORD"要

シューティングゲーム作成講座(4)

Sakamaki Katumi
坂巻 克巳

この連載では,シューティングゲームを作ることが最終目的です。しかし,具体的にどのようなゲームを作るかまでは,まだ決まっていません。ここでゲームデザインについて考えてみましょう。

今月も上杉氏はとんでもなく忙しく,とてもじゃないけど原稿を書く暇がないとのこと。そこで,シューティングゲームシステムのいい出しっぺの私,坂巻が急遽引っぱり出されてしまいました。

プログラム的な話は,上杉氏が具体的なプランをもっていることでしょう。ですから,私が口をはさむ必要はありません。ということで今回は,シューティングゲーム作成講座の番外編っぽい感じで,主にゲームデザイン関係を中心に話を進めていくことにします。

シューティングゲーム

まずは,基本的なことから始めましょう。シューティングゲームとはいったいどういったものなのでしょうか? なんて,改めて聞かれると返答に困ってしまうかもしれません。

まあ,ほとんどの人は,プレイヤーが操作する自機(飛行機でもワイバーンでもよし)がいて,ジョイスティックのボタンを押すと弾が出て,迫りくる敵を倒していく,そんなイメージを描いたことでしょう。

もしも,具体的なイメージが湧かない,シューティングゲームってなに? などと思ってしまった人は,ゲームのパッケージに「シューティングゲーム」と書かれているものを,片っ端から遊んでみてください。しかし,この連載を読んでいるということは,多少なりともゲームを作りたいと思っているはずですから,無用な心配かもしれないですけど。

ま,要するにただ単に敵が出てきて,それをプレイヤーが撃ち落とすだけでもシューティングゲームといえます。

しかし,それだけではつまらないですよね。なんでもいいのですが,現在,世に発表されているシューティングゲームをちょっと遊んでみればわかるとおり,ただ単に反射神経を駆使するだけのゲームといえど,実にさまざまな工夫がなされています。

そのなかでもいちばんわかりやすいのがグラフィックでしょう。でも視覚的な話は,S-OSの世界ではあまり関係ないと割り切って,もう少しシステムに目を向けてみると,いろいろなことが見えてきます。特に,現在のシューティングゲームがどのような要素を含んでいるかをよく観察してみるのです。

もちろん,ただ漠然と見ているだけでは意味がありません。見るべきポイントはいくつかあります。それは,自機の操作(パワーアップシステムも含む),敵のパターン(キャラクターとその動き),背景の3つです。この3つを理解すれば,そのゲームのデザインが見えてきます。

「シューティングゲームなんだから,自機と敵のパターンに注目するのは当たり前。でも背景にまで注目すべきなのだろうか?」もしかしたら,こう思った人もいるかもしれません。でも,シューティングゲームの背景だって,ただスクロールするために存在しているわけではないのです。

背景だって立派なキャラクターです。きちんとした意味をもたせれば,プレイヤーにメッセージを送ることもできます。

さらに面ごとのつながりを感じさせるように背景を構成することで,自然とプレイヤーにバックグラウンドストーリーを見せることだってできるのです。

言葉だとチープになりがちなものでも,視覚へ直接訴えるようにすることで説得力が増します。できることなら,ただ単に背景のバリエーションを増やすのではなく,見せることまで考えたいものです。

結局,ゲームに登場するすべてのものを使い,徹底的にゲームを見せるように心がけることで,ゲームの面白さが全然違ってきます。

そしてゲームデザイン

以上のことを踏まえて,頭の中でいろいろ考えていくと,おぼろげながら自分の作りたいゲームが思い浮かんできます。

思いっきり視覚的に派手なものも結構なのですが,作ろうとしているゲームはS-OS上で動くものです。おのずと表現力と処理能力に限界が見えます。

しかも,今回の場合はYGCSというシステムを利用するという制限が加わります(本来ならば,ゲームデザインを書き起こしてそれに合わせたシステムを制作するのが筋ですけど)。

しかし,実際のところYGCSによる制限はさして気になるほどのものではないでしょう。YGCSの仕様を見ればわかるとおり,ひとつひとつのルーチンはとてもシンプルなものです。逆に「これだけのルーチンでどうやってゲームを作るんだ」と思う人もいるかもしれません。

まあ,このあたりの話は,上杉氏が進めているところなので特に突っ込んだ話はしません。連載のこれからに期待しましょう。YGCSでなにか聞きたいこと,わからないことがあれば遠慮なく質問をOh!X編集部

までお寄せくださいね。

それでは少し話がそれてしまいましたが，YGCSを使ったシューティングゲームのデザインを考えていきましょう。

まずYGCSでは，いまのところ強制縦スクロールが基本なので，必然的に作るものは縦スクロールシューティングゲームとなります。そして，自機はパワーアップなしのノーマルショットのみ。速度アップアイテムもなし。でもダメージ＋残機制にしたいところ。

ステージは5面ぐらいで構成され，それぞれのステージにはボスキャラを用意します。敵のキャラクターは全部で15種類ぐらい。余裕があれば，それぞれのキャラクターに別パターンの動きを2，3種類用意しましょう。

だいたいこれぐらいの仕様が妥当なところだと思います。

ちなみに自機のパワーアップ，速度アップアイテムなしは，決して手抜きではありません。手抜きというよりもあとあとの混乱を防ぐための予防策とでもいっておくべきでしょうか。これは，下手にパワーアップシステムをつけてしまうと，ゲームバランスの調整のときに非常に面倒な結果を招くからです。

自機の武器が強すぎてはゲームが簡単になりすぎるだろうし，弱すぎては難しくなってしまいます。かといって逆に敵の強さを上げてしまうと，今度はプレイヤーがミスをしたときに復活することができなくなってしまうことになりかねません。まさにあちらが立てばこちらが立たず状態です。

その点，自機の強さが一定であれば，バランス調整は敵のパターンをいじるだけですみます。しかし，ゲーム全体のバランスをとるために，敵のパターンをよく練らないと，単純で面白くなく，しかもただ単に難しいだけのゲームになってしまうので注意が必要ですが。

あと，ダメージ＋残機制にする理由は，結局，自機はキャラクタ単位で移動を行うため，細かな操作，的確に敵弾をかわすことが難しいと思われるからです。慣れればこういった操作ミスを招くことがなくなるはずですが，それまでプレイヤーに我慢を強いるのは問題があるでしょう。あくまでもプレイしていて感じる爽快感を優先することにします。

また，先ほどもいったとおり，背景も積極的に活用したいものです。背景に紛れる敵キャラクターのトラップも用意するといいでしょう。

ステージデータ作成法

今度は，ステージデータを作成するうえでの注意点をいくつか書きましょう。

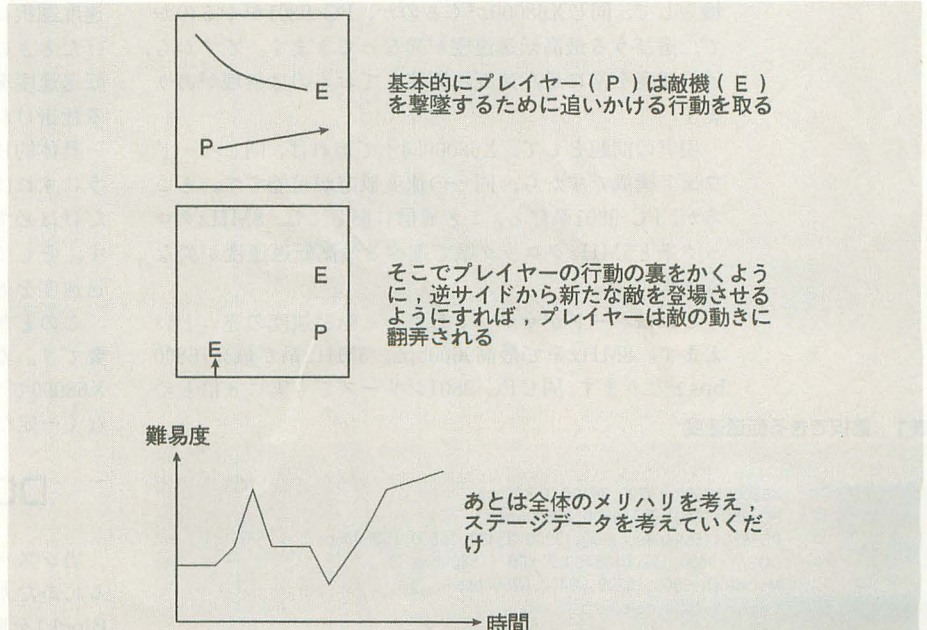
まず，メモリは有限です。そのため，できるかぎりデータを使い回すようにしなくてはなりません。

確かに多種多様な敵キャラクターたちを登場させることで，ゲームは派手になります。しかし，キャラクターが多くなるということは，必然的に使用されるメモリ容量が大きくなります。そしてなによりも雑誌に掲載する場合，あまり大きなものは許されません。

この問題の解決策として，ある一定の間隔を置いて再び同じキャラクターを登場させる方法があります（キャラクターデータと動きのデータを使い回せる）。

この方法のよい点は，以前登場したことのある敵ならば，動きを読めるのでプレイヤーはスムーズにゲームを楽しむことができます。ただし，あまり使い回しすぎるとゲームを遊ぶことが作業となってしまい，面白さが激減してしまうので注意しなくてはいけません。

また，同じキャラクターで動きだけが違う敵を用意するのも有効な手段です（キャラクターデータを使い回せる）。この方法を使うとデータの簡略化とともに，ゲームの難易度の高低の波も作れます。

基本的にプレイヤーは以前に登場した敵と同じキャラクターであれば，敵の動きのパターンを予測して迎撃しようとします。

しかし，動きが違うので当然プレイヤーの予想外のところに敵キャラクターは移動します。すると，プレイヤーは敵キャラクターを迎え撃つのではなく，追い掛けることになります。そして，いいタイミングで次の敵キャラクターを登場させると，結果的にプレイヤーは翻弄されることになります（図）。

ステージデータを作成するときには，このようにプレイヤーの意表をつく，つまりゲームにメリハリがつくような構成にすることが大切です。いつまでもだらだらと同じような難易度でゲームが続いていたのでは，すぐに飽きてしまいますからね。

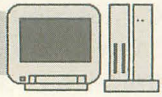
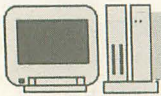
さらに，プレイヤーの行動を予測し，ときには思いどおりに，ときにはトラップに引っ掛かるように誘導してやることができれば完璧です。

そして，少ないメモリで最大の効果を得るように敵のパターンを検討すれば，きっと面白いゲームになるはずです。

こだわりのゲーム作り

以上，非常に個人的な見解をつらつら述べてきました。ゲームの作り方としては正しいのかどうか，私にはわかりませんが，やはり，練り込まれたゲームは面白いものです。面白いものを作ろうとするならば，中途半端で投げ出さず，徹底的にこだわってゲームを制作してみましょう。

もしも，満足できるものができなかったら，次の作品に向けてがんばればいいのですから。

図

仮想ドライバの開発実験PART6.

仮想ドライバの改良 その2

電机本舗 由井 清人 Yui Kiyoto

予告どおり今月はDOSのバージョンによる不具合を取り除いていきます。これで，マシン同士が理想的な転送速度で動作するようになります。そして，高速通信へのアプローチのため，SCC自体についても解説します。

今回も，7月号に掲載した仮想ドライバを小改良してみます。

現在ある問題点として，DOSのバージョンによっては当システムが作動しない可能性があります。これは，DOS自身がバージョンアップを繰り返しており，細かい仕様が変化しているということです。もしも，仮想ドライバが使用しているDOSの機能がDOSのバージョンによって変化していれば，障害が発生するはずですし，実際に起きています。

DOSのバージョンごとに機能が変化するのであれば，対応策として，仮想ドライバの従機が起動したときに，現在動いているマシンのバージョンを調べ，その結果により動作条件を変えてやればよいことになります。いずれにせよ，この場合は，DOSのバージョンごとに所定の動作を用意しておく必要があります。

そして，これが今回の最大の課題なのですが，転送速度の高速化です。転送速度の向上はそれだけでも技術的な問題を含んでいます。

しかし，それ以前にローテクノロジーな側面で，起動時に通信速度を変更しなくてはいけないという問題をもっています。はじめから，通信速度を最高に固定できれば話は簡単なのです。しかし，主機(X68000)の相手に従機として，同じX68000がくるのか，PC-9801がくるのかで，選びうる最高転送速度が異なってきます。ですから，通信速度を常に最高速度に固定しておくのは無理があります。

現実の問題として，X68000同士であれば，同じハードウェア構成ですから，同一の速度設定が可能です。ところが，PC-9801系だと，こと通信に関しては，8MHzクロック系と5MHzクロック系で選べる最高転送速度が異なります。

これはハードがサポートしている転送速度の違いといえます。8MHz系で最高9600bps，5MHz系で最高76800bpsとなります。同じPC-9801シリーズでも実に8倍もの速度の開きがあることになります。また，DOS/Vに関しては，最高で115200bpsですがX68000との組み合わせで実際に使用できるのは38400bps止まりとなります。次に各機種で選択できる転送速度を示します（表1）。9600bps未満は同じなのでリストには載せません。9600bps以上のみ示します。

通信は，常に双方が同じ転送速度でないと機能しません。ですから，X68000を軸にして，相手が選びうる最高速度しか選択できません。ですから，相手が5MHz系PC-9801ですと，転送速度の並びがX68000と同一ですから，X68000の上限76800bpsまで設定できます。これが8MHz系PC-9801ですと，最高転送速度は20800bpsですが，X68000で20800bpsという変則的なスピードはサポートしていないので9600bps止まりとなります。DOS/Vでも同様に転送速度を115200bpsまでサポートしていますが，X68000とランデブーできるのは38400bpsまでになってしまうのです。

参考程度に，Macintoshも載せておきました。この機種はDOS/Vとほとんど同じ並びであることに気がつきます。8MHz系PC-9801を除き，国産系とUS系でほぼ同じ速度体系をもっているのは興味深いかぎりです。

さて，このような具合ですので，仮想ドライブの転送速度選択にはひと工夫が必要です。主機と従機を立ち上げたときには統一した速度で通信を行って相互に最高の転送速度を通知し合い，選びうる最高転送速度へ移行する仕掛けが必要になってきます。

具体的には，主機，従機ともに9600bpsで立ち上げるようにすればよいでしょう。通常，すべての機種は9600bpsだけは必ずサポートしていますから，この速度が無難です。そして，次に従機側より主機へ，選択可能な最高転送速度を通知するようにします。

このとき，逆方向すなわち主機から従機への通知は不要です。なぜなら，このシステムの場合，主機は常にX68000ですから，選択できる転送速度は通知するまでもなく一定だからです（図1）。

表1　選択できる転送速度

```
X68000=9600,19200,38400,76800bps
PC-9801(8MHz系)=9600,20800bps
PC-9801(5MHz系)=9600,19200,38400,76800,153600bps
DOS/V=9600,19200,38400,57600,115200bps
Macintosh=9600,19200,38400,57600bps
```

DOSバージョン変化とその対策

当システムは仮想ドライブとして，ディスクを登録するにあたり，従機の各ディスクのDPB(Disk Parameter Block)を取得します。そして，ここに登録されている各

種情報を主機に登録し，仮想ドライブシステムを構築しています。

このDPBの取得はX68000のDOSコールGETDPB()にて行っています。PC-9801においては，同名の互換プログラムを作成して，これによりDPBを取得しています。PC-9801におけるGETDPB()をリスト1に示します。

このGETDPB()はMS-DOSのソフトウェア割り込み21番を内部で発行しています。ソフトウェア割り込みというのは，割り込みという名称がついていますが，特殊なサブルーチンコールと思ってください。割り込み番号21と，i8086のahレジスタにファンクション番号32を設定することによりMS-DOSにDPBテーブルの取得をリクエストします。

さて，GETDPB()はその第2パラメータ「struct DPBPTR *d」にDPBをセットする機能をもっています。ここで，問題が発生します。それはDPBテーブルの並びがDOSのバージョンにより変化しているということです。リスト2に各バージョンのDPBの並びを構造体定義したものを示します。比べてみてください。リスト中，メンバ名の頭に<使用>と書かれているのは当システムで参照する必須データです。

このリスト2のa）とb）を比べてみるとわかるのですが，最後の2つのメンバが異なります。差分を抽出してみましょう。

a） DOS Ver2.11の場合

```
/* カレントディレクトリのクラスタ番号　　*/
UWORD　dirfat;
/* カレントディレクトリ名　　　　*/
UBYTE　dirbuf[64];
```

b） DOS Ver3.XXの場合

```
/* 最後に変更したクラスタ番号　*/
UWORD　lastfat;
/* 未使用のクラスタ数　*/
UWORD　unusefat;
```

外見上は，DOS Ver2.11系と，3.XX系は異なっています。しかし，この異なるメンバは当システムでは使用しないので問題ありません。使用している情報に関しては完全に互換性があるといえます。もし，注意しないといけないとすれば，DPBテーブルの大きさが異なる点でしょう。ただ，これは，DOS Ver3.XXになってテーブルサイズは小さくなっているので問題はありません。

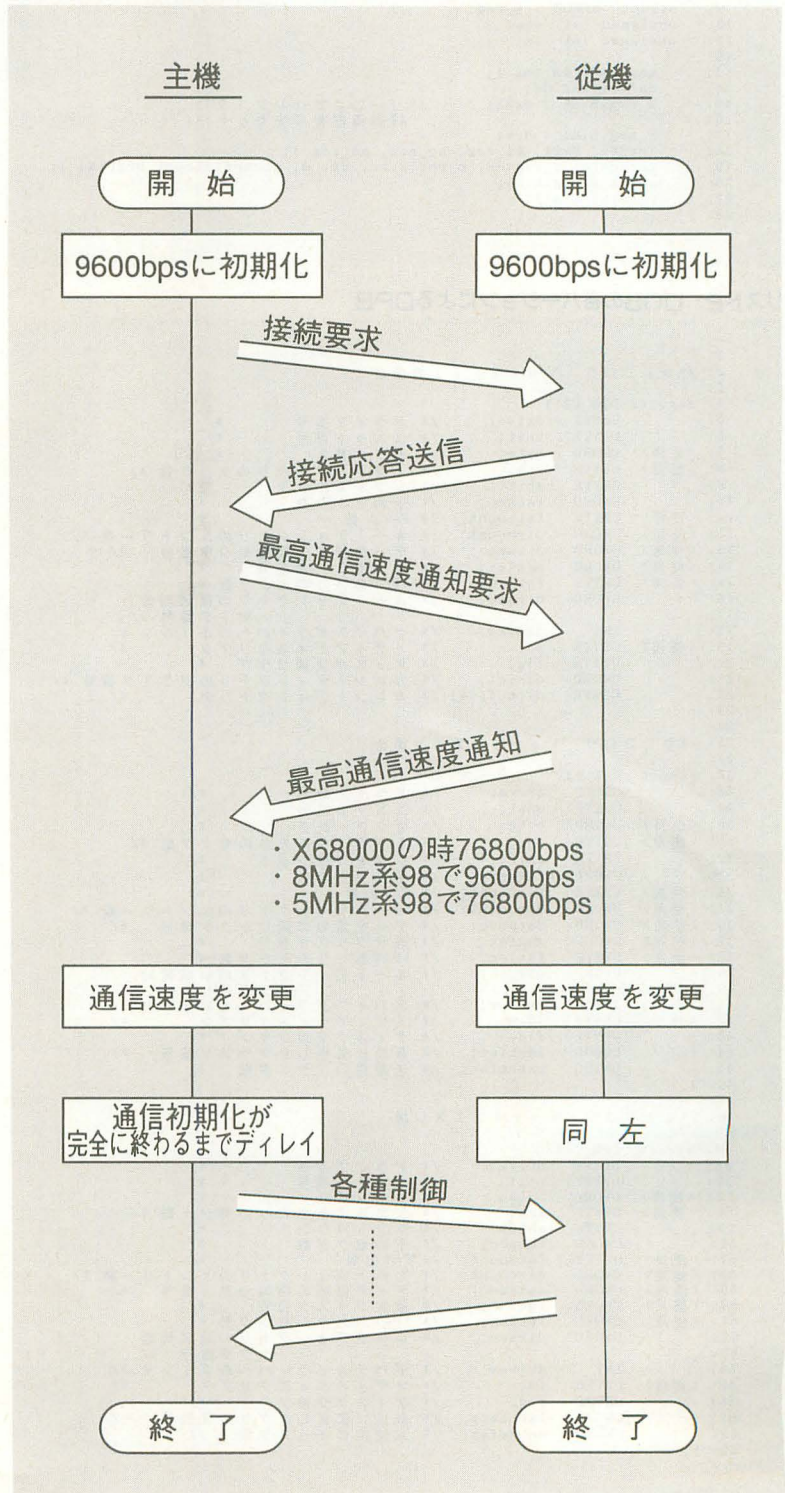
次は，b)とc)の相違です。見た目には両者は同一です。しかし，次のメンバをよく比較してください。

b） DOS Ver3.XXの場合

```
/* 1FATあたりのセクタ数 */
<使用>　UBYTE　fatlen;
/* ルートディレクトリの領域開始セクタ番号 */
　　UWORD　dirsec;
/* デバイスドライバへのポインタ */
　　INT　　driver;
/* メディアディスクリプタ　　　*/
<使用>　UBYTE　id;
/* 1FATあたりのセクタ数 */
　　UBYTE　fatlen;
```

c） DOS Ver4.XX以降の場合

```
/* 1FATあたりのセクタ数 */
<使用>　UWORD　fatlen;
/* ルートディレクトリの領域開始セクタ番号 */
　　UWORD　dirsec;
/* デバイスドライバへのポインタ */
　　INT　　driver;
/* メディアディスクリプタ　　　*/
<使用>　UBYTE　id;
```

図1　速度設定ネゴのバトン

/＊ 1FATあたりのセクタ数 ＊/

UBYTE　fatlen;

メンバ名，fatlenがVer3.XXまで1バイト型だったのがVer4.XXから2バイト型に変化しています。さらに問題なのは，fatlenの後方に位置するメンバidもこのシステムの必須参照データであるということです。ですから，fatlenは型が当然異なるので，Ver4.XXを境にした以前，以降で参照方法を変えないといけません。

また，メンバidにしたところで，テーブル中での配置が，fatlenのサイズが1バイト増えた分だけ後ろに1バイト圧迫されてずれています。このDOSの仕様の変更はハードディスクの大容量化により，引き起こされたものではないかと思います。1バイト型が2バイトになったわけですから単純に考えて容量は256倍まで拡大できる勘定になります。厳密には，FAT(File Allocation Table)つまりファイル管理領域の拡大ということでしょうか。1クラスタ当たりのサイズを大きくして大容量ディスクをサポートするのではなく，ディスクの管理領域の拡大により無駄のないファイル管理を実現するのが目的なのでしょう。いずれにせよ，当システムにおいてDOSのバージョンをチェックしなくてはいけない理由が出てくるわけです。

リスト3に簡単なDOSのバージョンチェックプログラムを示します。これは，非常に簡単なプログラムです。DOSの割込番号21，機能番号30のDOSのバージョン取得機能を内部でコールしています。戻り値として，第1パラメータのverにDOSのバージョンを，sub_verに小数点以下のバージョン番号を設定しています。ですから，DOS Ver3.11であれば，

ver＝3

sub_ver＝11

というように設定されます。あとは，この関数を，仮想ドライブシステムの中に組み込んでやればよいわけです。そして，取得したDOSのバージョンによりDPBテーブルの生成方法を選択して，主機であるX68000に正しいDPBを送るようにすればよいわけです。

ちなみに，従来の仮想ドライブシステムはDOS Ver2.XXおよび3.XXに合わせて作ってありました。

リスト1　GETDPB()

```
 1:
 2: /**************************************************/
 3: /*      GETDPB  DPBテーブル取得ルーチン          */
 4: /**************************************************/
 5: int     GETDPB( int drv, struct DPBPTR *d )
 6: {
 7:   union REGS    i_reg;
 8:   union REGS    o_reg;
 9:   struct  SREGS   s_reg;
10:   unsigned int  sts;
11:   unsigned int  ds;
12:
13:     segread( &s_reg );
14:     ds = s_reg.ds;
15:     i_reg.h.ah = 0x32;        /* <--ここでahレジスタに
16:                                   32の機能番号をセット */
17:     i_reg.h.dl = drv;
18:     int86x( 0x21, &i_reg, &o_reg, &s_reg );
19:     movedata( s_reg.ds, o_reg.x.bx, ds, d, sizeof(struct DPBPTR) );
20:     sts = o_reg.h.al;
21:     return( sts );
22: }
```

リスト2　DOSの各バージョンによるDPB

```
 1:
 2: /*a) DOS  Ver2.11の場合
 3:
 4: struct  DPB_V211        {
 5:         UBYTE   drive;      /* ドライブ番号         */
 6:         UBYTE   unit;       /* ユニット番号         */
 7: <使用>  UWORD   byte;       /* セクタ長さ           */
 8: <使用>  UBYTE   sec;        /* 1クラスタあたりのセクタ数 */
 9:         UBYTE   shift;      /* シフトカウンタ         */
10:         UWORD   fatsec;     /* 予約セクタ数           */
11: <使用>  UBYTE   fatcount;   /* FATの数                */
12: <使用>  UWORD   dircount;   /* ルートディレクトリのエントリー数 */
13: <使用>  UWORD   datasec;    /* データ領域の開始セクタ番号  */
14: <使用>  UWORD   maxfat;     /* 最大クラスタ番号       */
15: <使用>  UBYTE   fatlen;     /* 1FATあたりのセクタ数 */
16:         UWORD   dirsec;     /* ルートディレクトリの領域開始
17:                                               セクタ番号 */
18:         INT     driver;     /* デバイスドライバへのポインタ */
19: <使用>  UBYTE   id;         /* メディアディスクリプタ      */
20:         UBYTE   flg;        /* ディスク交換フラグ   */
21:         UWORD   dirfat;     /* カレントディレクトリのクラスタ番号 */
22:         UBYTE   dirbuf[64];/* カレントディレクトリ名       */
23: };
24:
25: /*b) DOS  Ver3.XXの場合
26:
27: struct  DPB_V3XX        {
28:         UBYTE   drive;      /* ドライブ番号           */
29:         UBYTE   unit;       /* ユニット番号           */
30: <使用>  UWORD   byte;       /* セクター長さ           */
31: <使用>  UBYTE   sec;        /* 1クラスタあたりのセクタ数 */
32:         UBYTE   shift;      /* シフトカウンタ         */
33:         UWORD   fatsec;     /* 予約セクタ数           */
34: <使用>  UBYTE   fatcount;   /* FATの数                */
35: <使用>  UWORD   dircount;   /* ルートディレクトリのエントリー数 */
36: <使用>  UWORD   datasec;    /* データ領域の開始セクタ番号  */
37: <使用>  UWORD   maxfat;     /* 最大クラスタ番号       */
38: <使用>  UBYTE   fatlen;     /* 1FATあたりのセクタ数 */
39:         UWORD   dirsec;     /* ルートディレクトリの領域開始
40:                                           セクタ番号  */
41:         INT     driver;     /* デバイスドライバへのポインタ */
42: <使用>  UBYTE   id;         /* メディアディスクリプタ      */
43:         UBYTE   flg;        /* ディスク交換フラグ   */
44:         UWORD   lastfat;    /* 最後に変更したクラスタ番号  */
45:         UWORD   unusefat;   /* 未使用のクラスタ数   */
46: };
47:
48: /*c) DOS  Ver4.XX以降
49:
50: struct  DPB_V4XX        {
51:         UBYTE   drive;      /* ドライブ番号           */
52:         UBYTE   unit;       /* ユニット番号           */
53: <使用>  UWORD   byte;       /* セクタ長さ             */
54: <使用>  UBYTE   sec;        /* 1クラスタあたりのセクタ数 */
55:         UBYTE   shift;      /* シフトカウンタ         */
56:         UWORD   fatsec;     /* 予約セクタ数           */
57: <使用>  UBYTE   fatcount;   /* FATの数                */
58: <使用>  UWORD   dircount;   /* ルートディレクトリのエントリー数 */
59: <使用>  UWORD   datasec;    /* データ領域の開始セクタ番号  */
60: <使用>  UWORD   maxfat;     /* 最大クラスタ番号       */
61: <使用>  UWORD   fatlen;     /* 1FATあたりのセクタ数 */
62:         UWORD   dirsec;     /* ルートディレクトリの領域開始
63:                                           セクタ番号  */
64:         INT     driver;     /* デバイスドライバへのポインタ */
65: <使用>  UBYTE   id;         /* メディアディスクリプタ      */
66:         UBYTE   flg;        /* ディスク交換フラグ   */
67:         UWORD   lastfat;    /* 最後に変更したクラスタ番号  */
68:         UWORD   unusefat;   /* 未使用のクラスタ数   */
69: };
```

高速通信へのアプローチ

X68000には通信用LSIにザイログ社のZ8530SCC（以後SCCと称する）が使用されています。この石は，もともとは同じくザイログ社の16ビットCPUであるZ8000用に開発されたものです。内部にA，Bの2つの通信チャンネルをもっています。X68000ではBチャンネルをマウス専用に使用しています。ですから，我々が通常使うRS-232Cには，SCCのうちのAチャンネルを割り振り，ここ

リスト3　DOSのバージョンチェックプログラム

```
 1:
 2: /**************************************************/
 3: /*      getver  DOS Version 取得ルーチン          */
 4: /**************************************************/
 5: void getver( int *ver, int *sub_ver )
 6: {
 7:   union REGS      i_reg;
 8:   union REGS      o_reg;
 9:
10:         i_reg.h.ah = 0x30;
11:         i_reg.h.al = 0;
12:         int86x( 0x21, &i_reg, &o_reg, &s_reg );
13:         *ver = o_reg.h.al;
14:         *sub_ver = o_reg.h.ah;
15: }
```

で通信制御が行われることになります。

通常の通信制御ですと，IOCSなりDOSコールなり，Human68kがアプリケーションに提供する通信機能を利用して行います。ですが，通信最大速度はいいところで19200bpsまでです。これ以上の速度はHuman68kではサポートされていません。また，Human68kによる機能提供は通常割り込み処理により行われます。

この割り込みは，外部からRS-232Cに入力が起きると，まずSCCが1バイトの受信処理を行います。そして，受信処理を終えたときにCPUに割り込みと呼ばれる信号を送ります。

CPUはこの割り込み信号を受けると，現在行っていた処理（通常のアプリケーションやHuman68kなどのなんらかのプログラム）を中断して，RS-232Cの受信を行います。受信処理は通常はSCCからデータを受け取りメモリ上のRS-232C受信バッファに格納することです。この処理そのものも，実はシステムにサブルーチンとして格納されています。いい変えると，CPUはSCCから割り込み信号を受けると現在実行中のプログラムを中断し，RS-232C受信用のプログラムを実行するわけです。そして，これが終わると，なにくわぬ顔をして，中断していた元のプログラムの処理を再開します。

この通常行われている処理では，プログラムがなにも考えなくても，いつのまにか通信処理をしてくれるので楽です。アプリケーションは通信データがあるかどうか，定期的に通信バッファをチェックし，もしデータが格納されていればIOCS（もしくはDOS）コールにより，データをバッファより引き落とせばよいわけです。

割り込みによる通信制御の不都合

では，この割り込みによる制御はいいことずくめなのでしょうか。受信が発生するたびに割り込みが発生して，CPUは制御をSCCへ移し，トンボ返りしてきます。この工程が，瞬く間でタイムロスがゼロであれば問題ありませんが，やはりロスタイムは発生します。

また，通信速度が低速であれば，次の受信データをSCCが1ビットずつ受信している間に，CPUはSCCより受領すればよいわけです。

しかし，通信速度を速くしていくと，CPUはついてこられなくなります。割り込みが発生し，SCCへデータを取りにきたときには，すでに次の受信データが追い越してきているという事態が発生します。つまり，この場合にはデータの取りこぼしが発生します。こぼしたデータは，次からきたデータに上書きされるために消えてなくなってしまうのです。

ここに，割り込みによらない通信の必要性があるのです。だいたい，割り込みによる通信速度の上限は，IOCSが用意している通信速度くらいまでと思っていてください。19200bps程度までがストレスのない速度でしょう。これ以上の速度を出したいのであれば，ひと工夫が必要となります。

19200bpsという速度は，理論値で計算したときに，bps (bit par second) すなわち1秒間に送出できるビット数です。これより，1ビット送出するのに要する時間は，1/19200＝0.000052秒となります。そして，1バイト（8ビットのデータ）ですが通信においては通常，スタートビットとストップビットの2ビットを加算しますから10ビットとしてSCCは処理します。

これより，1バイトのデータを送るのに，かかる時間は先ほどの0.000052秒×10＝0.00052秒。つまり520μsとなります。19200bpsの速度のときにデータが隙間なく，詰め詰めで連続送信されたときには520μsごとに次のデータがくることになります。少なくとも，割り込み処理はこの時間内に全処理を終了しないと，次のデータを取りこぼすことになるのです。そして，今回実験を考えているのは，オーバー38400bps，あわよくば76800bpsですから時間間隔は，260μs (38400bps) ないし130μs (76800bps) の間に，全割り込みの処理をしなくてはいけないことになります。

この速度がCPUにとって速いのか，それともたいしたことがないものなのか読者の皆さんにはわからないと思います。ここで，CPUのクロック速度と比較してみましょう。たとえば，初代X68000のクロック速度は10MHzでした。この速度は，1秒間に10M（メガ）サイクル信号を発生するものです。メガは1,000,000ですから10M＝10,000,000サイクルということになります。

以上のことから，クロックが1サイクル発生する時間間隔は，

1秒／10,000,000

すなわち，0.1μsということになります。また，

130μs　：　0.1μs
（割り込み処理遂行時間）　（CPUの実行最小単位）

という図式が発生してきます。これは，意外と大きな比率と思うかもしれません。もしもCPUが機械語命令を1

表2　内蔵の通信チャンネルの機能

＊非同期モード
- 5，6，7，8ビット/キャラクタ
- 1，1.5，2ストップビット/キャラクタ
- 偶数パリティ，奇数パリティ，パリティなし
- x1，x16，x32，x64クロックモード
- ブロックの生成と検出
- パリティ，オーバーラン，フレーミングの各エラーの検出

＊同期モード
- バイト指向同期モード
  - キャラクタ同期は，内部，外部のいずれも可能
  - 1または2個の同期キャラクタ
  - 同期キャラクタは6または8ビット
  - 同期キャラクタの自動挿入または削除
  - CRCの生成と照合
- SDLC/HDLCモード
  - アボートシーケンスの生成と検出
  - 自動ゼロ挿入と削除
  - メッセージ間での自動フラグ挿入
  - アドレスフィールドの検出
  - 情報フィールドの端数処理
  - CRCの生成と照合
  - SDLCループモードのEOP検出によるエントリ（オンループ）と脱出
- データ転送速度
  - 最大1.5Mbits/sec（モノシンク，バイシンク）
  - 最大375Kbits/sec（FM符号化方式DPLL）
  - 最大187Kbits/sec（NZRI符号化方式DPLL）

つ実行するのに，平均10クロック（命令により必要とするクロック数は異なる。もっとも最新のCPUは1命令1クロックになりつつある）かかるとすれば，機械語命令に換算して割り込みプログラムは機械語にして130命令以内に収めないとタイムオーバーを起こすことになる数字だと思ってください。

このようなことより，割り込みによる制御ではなく，別のもっと処理時間に有利な方法が必要になってくるわけです。

実際にはアプリケーション自身が直接SCCを制御し，直接SCCを制御すればよいわけです。アプリケーションの中で常時SCCを監視し，もし，受信データが発生したらアプリケーションが即座にデータを読めばよいわけです。このような通信制御をポーリングによる通信制御と呼びます。

表3　制御レジスタ

・読み込みレジスタ

レジスタ	内容
RR0	送信/受信バッファステータスおよび外部ステータス
RR1	特別受信条件ステータス，端数コード，エラー条件
RR2	修飾割り込みベクタ（チャンネルBのみ） 非修飾割り込みベクタ（チャンネルAのみ）
RR3	割り込み保留ビット（チャンネルAのみ）
RR8	受信バッファ
RR10	雑ステータス，ループ/クロック情報
RR12	ボーレートジェネレータの時定数　下位バイト
RR13	ボーレートジェネレータの時定数　上位バイト
RR15	外部/ステータス割り込み情報

＊両チャンネル共通のRR2レジスタは，ほかの7本のレジスタと区別すること

・書き込みレジスタ

レジスタ	内容
WR0	各種モードのCRCの初期化および初期設定コマンド。レジスタポインタ
WR1	送信/受信割り込みおよびデータ転送モードの定義
WR2	割り込みベクタ（各チャンネルからアクセス）
WR3	受信パラメータおよび制御
WR4	受信/受信雑パラメータおよびモード
WR5	送信パラメータおよび制御
WR6	同期キャラクタまたはSDLCアドレスフィールド
WR7	同期キャラクタまたはSDLCフラグ
WR8	送信バッファ
WR9	マスタ割り込み制御およびリセット（各チャンネルからアクセス）
WR10	雑トランスミッタ/レシーバ制御ビット
WR11	クロックモード制御
WR12	ボーレートジェネレータの時定数　下位バイト
WR13	ボーレートジェネレータの時定数　上位バイト
WR14	各種の制御ビット
WR15	外部/ステータス割り込み制御

＊両チャンネル共通のWR2，WR3レジスタは，ほかの14本のレジスタと区別すること

図2　SCC周りのダイアグラム

SCCの構造と制御方法

ポーリング制御を行うにあたり，SCCの内部構造をよく理解しておく必要があります。

内蔵の通信チャンネルA，Bは表2の機能をサポートしています。表中の同期モードとSDLC/HDLCモードは通常我々には関係ないでしょう。通常，および今回使用する非同期モードと呼ばれる機能です。

また，機構的には，図2のブロックダイアグラムにより構成されます。概略的には3つの部分で構成されていて，CPUとのインタフェイス，発信機，そして並列直列コンバータより構成されます。X68000のRS-232Cコネクタには，外部インタフェイスと結合（厳密には直結ではなく，バッファと呼ばれる一種の出力増幅装置を経由される）されています。

SCCは，CPUから受け取った1バイト並列データを発信機の生成するクロックに合わせて1ビットずつ外部へ送出して，並列データを直列データに変換しています。

さらにSCC自身は内部に，通信制御するための表3の制御レジスタをもっています。各チャンネルごとに書き込みレジスタ14個，読み込みレジスタ7個，そして両チャンネル共用が2個です。CPUはこれらレジスタを操作し，SCCを制御して通信を行います。

Human68kがもっているIOCSコールによる通信機能も，実は内部の割り込みプログラムでこれを利用しているわけです。

次回について

今回は少し急いでSCC部分を説明してきたきらいがありますが，次回はさらに，SCCのポーリングによる通信制御の使い方を紹介し，実際にSCCの初期化，データ受信，送信の実験を行っていきます。そして，条件が整うならば仮想ドライブシステムへ組み込んでいきたいと思います。

おそらく仮想ドライブシステムのRS-232C入出力処理は，そのままポーリング型RS-232Cと換装できるのではないかと思います。ただ，従来使っている割り込み型制御特有の長所として，受信データの自動格納による取りこぼしの保護があります。具体的な恩恵としてLOF232C()という受信バッファチェック機能があります。これに対してポーリングでは，常時SCCを監視しなくてはいけないので，この機能を作ることができません。現在の仮想ドライブシステムはLOF232C()を使用していますので，制御構造を変更する必要がありそうです。このあたりが厄介といえます。

ぼちぼちと対策を考えていますが，どのようになるかは次回のお楽しみとしておきましょう。

SXコールを使ったファイル管理

Ishigami Tatsuya 石上 達也

付録のディスクのたびにどんどん周辺まで広がっていくSX-BASICワールド。処理系本体もSX-BASICもより安定したver.0.6をお届けします。新規命令はpath$だけですが，見違えるくらい使いやすくなっています。

今月のバグ出し

「もみじ狩りPRO-68K」に収録されたSX-BASICに対して以下のようなバグが見つかりました。

●SX-BASIC

・関数freadsの引数に256の整数倍の大きさを持つ文字列変数を指定すると，正しく文字が読み込まれない。

・256文字以上の文字列に対応していない関数があった

・引数として与えられた文字列変数sに対してs [n] が正しく値を返さない。

・ローカル変数宣言時にコメントを書くとエラーとなる。

例)　int i = 1　/* comment

・ダイレクトコマンド入力時に文字列を入力せずにリターンキーを数回押すとSX-BASICが暴走する。

・getmes()で解放していないメモリハンドルがあった。

・スクロールバーが1ドットずれている。

●ウィンドウエンジン

・ボリューム，アップダウンボタンなどのアイテムに対して，value/min/maxプロパティに負の値を代入すると，正しく動作をしない。

●ウィンドデザイナ

・配列アイテムを作成する際，最後の要素になるアイテムがうまく作成されない。

・シャーペンで作成されたテキストに対し，EdEVデータ(シャーペン独自のデータ)を保存しない。

*　　*　　*

この場を借りて，これらのバグを発見し，連絡してくださった方々に深くお礼を申しあげます。

昨日まで手元にあったバージョンではこれらのデバッグは完了していたのですが，いきなりディスクが飛んでしまったので，今回の付録ディスクではSX-BASIC本体のバグ取りをいくつかと若干の機能拡張を行ったものが収録されています。

なお，今回追加された機能は，

・path$

プログラムのあるパスをフルパスで返すというシステム変数です。

再コンパイルする方へ

SX-BASICは，SX-WINDOWがバージョン1.10だった頃に制作に取り組みました。当時は「SX-WINDOW開発キット」(以下「開発キット」)が正式リリースされる以前で，開発環境として参考文献1に収録のインクルードファイルとライブラリを用いていました。

その後しばらくして，インクルードファイルの内容が大幅に変更された純正の「開発キット」が発売されました。

SX-BASIC ver.0.5では「SX-WINDOW開発キットサポートツール集」(以下「ツール集」)に収録のインクルードファイルとライブラリを使用して，アコーディオンメニューやフロートウィンドウといった機能を実現しています。

「ツール集」は「開発キット」よりもあとに出たので，当然「開発キット」の存在を前提に設計されています。

レクタングルを表す構造体は，参考文献1では「rect」でしたが，「開発キット」では「Rect」のように頭文字が大文字になっています。「ツール集」でもこのようになっています。

本来なら，正式な「開発キット」が発表された時点で，SX-BASICもその流儀に則ったプログラムにすべきだったのですが，諸般の事情により，「ツール集」のインクルードファイルのほうを参考文献1の流儀にあわせたものを用意してそれを使っています。

ですから「もみじ狩りPRO-68K」に収録されたソースリストから実行ファイルを作成するには，

(標準の) CCompilerPRO-68K

(標準の) 開発キット (2種類のインクルードファイル群を収録していますが，このうちで過去のものとの互換性を保持している「sxdef.h」のほうです)

(上記の変更を加えた) ツール集のインクルードファイル

が必要となります。

その後いろいろ考えた結果，やはり「開発キット」の流儀でいこうとなりましたので，次回のバージョンからはこのような変更が必要なくなります。

ですから，どうしてもver.0.5を再コンパイルをするんだ，という方は上記の変更を加え (もちろん，変更前のファイルもバックアップしておきます)，そうでない方は「もみじ狩りPRO-68K」に収録のソースリストは参考程度にとどめ，そのまま再コンパイル可能なソースリストは次回の付録ディスクまでお待ちください。

このような事態になってしまったことを，この場を借りて深く反省いたします。

vanish

SX-BASICそのものは，X-BASICにほんの少しの拡張を加えた機能しか持ってい

ません。その代わり，SX-WINDOWという環境の中でほかのプログラムと密接に連絡を取りあい，共同してひとつのアプリケーションを構築するという手法が使えます。

ほかのアプリケーションと連絡をとる方法として，

起動時に必要な情報を引数として渡す

という方法があります。

fock()関数を用いると，

fock("sxplay *ooedo.opm*")

のように，ooedo.opmというファイルを演奏することができます。これは，SX-BASIC本体にはない音楽演奏機能を，SXPLAY.X(中森章氏作，本誌92年6月号付録ディスク「創刊X周年記念PRO-68K」に収録）というプログラムと協力することによって実現したものです。

さて，ここでひとつ問題があります。

音楽機能が必要なSX-BASICのプログラムは，当然，プログラムの先頭でSXPLAY.Xを実行します。65535色のグラフィック表示機能が必要なプログラムはSXPICS.X（石上達也「SX-WINDOW上のPICローダSX-PICSLICE」本誌1994年8月号）を実行します。

そして，SX-WINDOWを終了し，次回，SX-WINDOWを起動した場合を考えてください（「再起動」した場合も同様です）。

SX-WINDOWの起動と同時に，どのSX-BASICからも呼び出されていないSXPLAY.XやSXPICS.Xが起動されます。これは，SX-WINDOWが前回使用された状態を保存する機能があるからで，「終了」時に起動されていたタスクは，次回SX-WINDOWが起動されたときに，同じ位置にウィンドウを開いているのです（田村健人「SYSDTOP.SXを斬る」本誌1994年9月号参照）。

このため電源を切っても自分が作業しやすいように並べたウィンドウ，ファイルアイコンなどは，そのままの位置にあり，効率的に作業を進めることができるようになっています。

SX-BASICは自分で協力プログラムを起動することによってそのプログラムのタスクIDを知ることができました（fock()関数の戻り値として）。SX-WINDOWの再起動時にプログラムを起動するのはタスクマンと呼ばれるSX-WINDOWの一部分です。

SX-WINDOW起動時に自動的に起動されるタスクのタスクIDは，タスクマンが知っているだけで，SX-BASICには知らされません。プログラム名から強引にタスクIDを求めることもできますが，同じファイル名を持ったタスクが複数あった場合，どのタスクを協力プログラムとみなしてよいかSX-BASICには知る術がありません。

ですから，タスクマンが自動的に起動したプログラムとは別に，SX-BASICは新たに協力プログラムを起動しなければなりません。前者のプログラムは使われることがなく，ただ，メモリを無駄に占拠するだけです。

タスクマンが自動的にプログラムを起動するのを禁止するのが，SX-BASIC ver.0.5から追加されたvanish()関数です。このvanish()関数は引数としてタスクIDを指定すると，SX-WINDOW再起動時に自動的にそのタスクを起動するのを禁止します。

例）

```
int id;
id = fock("sxplay.x")
vanish(id)
```

以後は，SX-BASICからほかのプログラムを起動する場合は，なるべくvanish()関数を用いてください（この機能は，タスク管理テーブルのrsv1 [2] の17ビット目を1にするという未公開機能によって実現されています）。

今月のサンプルプログラム

最近のハードディスクの値下がりによって，なんでもかんでもファイルをハードディスク上に入れることができるようになり，4段や5段もあるような深いディレクトリを掘ることも珍しくなくなりました。

なにも考えずにこれをやると，普段からものをなくす癖のある人は，ハードディスク上でもファイルをなくすという癖がつくようになります。

コマンドライン上だと，whereコマンドを使って，どのディレクトリ上に保存していたかを調べることができるのですが，SX-WINDOWの標準アクセサリだけでは思い当たるディレクトリアイコンを片っ端からダブルクリックして回らなければなりません。

今回はこのような不便を解消するために，whereに相当するプログラムを作ってみます。

下準備

ファイルの管理は基本的にSX-WINDOWのタスクマンが行っているはずですから，ファイルの検索は自前で行わず，なるべくタスクマンに伺いをたてるようにすべきでしょう。

関数名 TSSearchFile
マネージャ名 タスクマン
SXコール番号 $A403
書式

```
#include <task.h>
long TSSearchFile(const char *sname, char *dname, const char *path);
/* const char *sname;    ファイル名
 * char *dname;
 *         該当ファイル名格納アドレス
 * const char *path; カレントパス */
```

機能

snameのファイルをpathのパスより検索し，見つからなかった場合はすべてのパスより探し見つかったフルパス名をdnameにセットします（ある程度検索したところで，ダイアログを出します）。TSSearch FileND関数と同様の処理を行いますが，検索中のダイアログを出します。

戻り値

ファイル長を返します。エラーの場合はエラーコード(負の数)を返します。

注意

再配置が発生します。

（ツール集に収録の「インサイドSX」より転載）

というのが，ファイル検索を行ううえで，有効な手段のようです（石上注：SX-WINDOW ver.3.0になってから，検索中のダイアログが表示されないようになったようです。検索のキャンセルにはブレイクキーを使ってください）。

ここで，char *～というのは，～という文字列が収められたアドレスを指定しなさい，という意味です。正直にこの規則に従うなら，SX-BASICの内部情報を収得しここに代入しなければなりません。しかし，そのような内部情報を習得するような命令

はありませんし，仮にあったとしても著しくSX-BASICのバージョンに依存するはずですので，あまり使用はおすすめできません。

では，どのようにアドレスを指定するのでしょうか？

関数名　MMChPtrNew
マネージャ名　メモリマン
SXコール番号　$A01E
書式

```
#include <sxmemory.h>
Pointer MMChPtrNew(long size);
    /* long size;ブロックサイズ */
```

機能

カレントヒープからsizeバイトのポインタ(再配置不能ブロック)を確保します。内部処理でMMPtrNew関数を呼び出しています。

戻り値

ポインタ(再配置不能ブロック)を返します。エラーの場合は NULL(0)を返し，そのエラーコードはMMMemErrorGet関数で調べることができます。

注意

再配置が発生します。sizeに負の値や，非常に大きな値を指定した場合の動作は保証されません(メモリマンは，指定されたサイズの正当性をチェックしていません)。

(インサイドSXより)

この機能を用いて，メモリ領域を確保します。その領域に文字列を転送してやれば，先頭アドレスをTSSearchFileに渡すことができます。

このように，メモリ空間の一部を専用の作業用に割り振った場合は，作業が終了して必要がなくなった時点で解放してやり，ほかの作業用に再び割り振れるようにしておくのがSX-WINDOW上でのエチケットです。

関数名　MMPtrDispose
マネージャ名　メモリマン
SXコール番号　$A02F
書式

```
#include <sxmemory.h>
void MMPtrDispose(Pointer ptr);
    /* Pointer ptr;    ポインタ */
```

機能

指定したポインタ(再配置不能ブロック)を解放します。この関数は例外的に，ptrにNULL(0)を指定することができ，その場合処理はなにも行われません。

戻り値

戻り値はありません。

注意

この関数実行後，ポインタ(再配置不能ブロック)を解放したことを明示するために，ptrにNULL(0)を代入してください。

(インサイドSXより)

最後の注意が少し気になるところですが，おそらくプログラマの不注意で同じポインタが2度破棄されないようにするためのアドバイスだと思われます(「機能」の項参照)。今回のプログラムではポインタを管理する変数はローカル変数で，ポインタの破棄後，そのポインタを管理する変数自体がなくなってしまうので，このようなこと柄は起こりえません。「注意」は無視することにします。

以上で，下準備は整いました。

使い方

以上のSXコールを用いてプログラムを

表1

TSSearchFile2	
書　式	#include <task.h> long TSSearchFiles2(int mode1, int mode2,void * sfile,const char * sname,char * dname,const char * path); /* int mode1;　検索回数 * int mode2;　検索モード * void * sfile;　ファイル検索レコードへのポインタ * const char * sname;　ファイル名へのポインタ * char * dname;　該当ファイル名格納アドレス * const char * path;　カレントパス */
機　能	snameで指定したファイルを検索してdnameに格納します(90バイト必要)。 検索する順番は，snameにパスが含まれる場合はそのパスを，以外はpathで指定されたパスを最初に探し，見つからなかったときはドライブAのルートより検索します。 ファイル検索レコードには268バイト必要です。 ワイルドカードを使用して複数のファイルを検索する場合は，同一のsfileをパラメータにして複数回呼び出します。 ファイル名は最初にワークにコピーされるので，格納バッファと同一でも構いません。 mode1が0以外のときは，指定した回数検索したところでダイアログを出します。
戻り値	ファイル長を返します。 エラーの場合はエラーコード(負の数)を返します。
注　意	再配置が発生します。
解　説	mode2で指定するモードによって処理が異なります。 mode2＝0:ひとつファイルを見つけて終了 最初のファイルが見つかるまですべてのドライブを捜します。 検索後ダイアログの廃棄などの終了処理を行います。 mode2＝1:複数検索するときの最初の検索 複数ファイル検索するときの最初の検索です。 ファイル検索レコードを初期化し，最初の検索をします。 ひとつファイルが見つかるか，1ドライブ探したところで処理を中断します。 処理をやめるときは必ずmode2＝3で終了します。 mode2＝2:複数検索するときの次の検索 mode2＝1のつづきの処理を行います。 mode2＝3:複数検索するときの終了処理 複数検索するときの終了処理です。 ダイアログの廃棄などを行います。 mode2＝4:ダイアログのアップデート 複数検索の途中でアップデートイベントが発生したときに，ダイアログのアップデートを行います。 ●発生するエラーコード ER_FILENOTFND (-0x1fe1)：ファイルが見つからない ER_SERCHBREAK (-0x1fe2)：ファイルの検索が中止された ER_SERCHONEDRV (-0x1fe3)：1ドライブ検索した ER_DRVNOTREADY (-0x1fe4)：ドライブの準備ができていない

図1：各アイテムの配置

組むわけですが，その前に仕様を決めておきましょう。

ウィンドウデザインは，図1のようにしました。

Text1にファイルネームを入力すると，Text2に検索結果を返します。このとき，検索開始を指示するための特殊なボタンなどは用意しません。ファイルネーム入力中にリターンキーが押されると，すぐさま検索を開始します。

プログラムについて

リスト1に示すプログラムが，今回制作したプログラムです。

1～14行目までが，配置するアイテムとそれに付随する関数です。関数といっても，Text2検索結果表示専用，Text3，Text4は，それこそ文字を表示しているだけ（「↓」，「検索するファイル名(ワイルドカード可)」）です。実際に，イベントに対する動作を記述しているのは，3～5行目の関数Text1_KeyInだけです。

15行目からが，実際にファイル検索を行う関数です。

順を追って説明していきましょう。

まず，17，18行目で90バイトの大きさを持つメモリポインタを確保しています。ひとつめのメモリポインタには検索するファイルの名前が，2つめのメモリポインタには，検索を始めるパスを指定することになっているのですが，今回は使用しないので，空文字列を代入します（23行)。

19行目で宣言されている文字列型変数のgetnameは検索された結果を代入するものです。

20行目からが，17行で確保したメモリポインタに文字列を転送するループです。本来なら，

asc(mid$(filename, i ＋ 1, 1))

は，

filename [i]

ですむはずですが，前述のSX-BASICのバグにより，今回はこのような回りくどい表現になっています。次期バージョンからは，後者の表現が使えるようになります。

それはさておき，役者が揃ったところで，いよいよ実際にファイルの検索を行います(24行)。

26行目から，実際にファイルが見つかったかどうかを確認し，見つかった場合には，29～32行のループで「そのファイルの存在するドライブ名」+「ディレクトリ名」+「ファイル名」をgetnameに取得します。31行も同様の理由により，次期バージョンからは，

getname [i] = peek(name ＋ i)

のように記述できるようになります。

34，35行目で使用済みのメモリポインタを破棄し，36行目でテキストアイテムText2に得られたファイル名を代入しています。

プログラムの改良（その1）

リスト1で，一応，当初の目標は実現で

ポインタとハンドル

いまさらいうまでもなく，アドレスを用いてメモリ上にあるデータを示すのがポインタで，そのポインタのアドレスを指し示すのがハンドルです。

C言語でいうと，

```
char ＊ ptr = "Hello World !!";
char ＊＊ hdl = &ptr;
```

前者がポインタ，後者がハンドルということになります。

SX-WINDOWでプログラミングを行っていると，ある特定のポインタやハンドルへ文字列を代入するようなサブルーチンを作りたいことがよくあります。

サブルーチンから見れば，「ほいっ」と素性のわからないポインタが渡されるわけです。ひょっとしたら，そのポインタはこれから書き込もうとしている文字列の大きさ（長さ）よりも小さな領域しか持っていないかもしれません。

SX-WINDOWのプログラミングを行う際には，このような処理を行う前に与えられたポインタやハンドルが十分な大きさを持っているか否かをチェックし，大きさが足りない場合には拡張してやる必要があります。

ここで，ちょっと考えてみれば不思議に感じると思うのですが，ポインタやハンドルの大きさというのは，いったいどうやって調べることができるのでしょうか。

SXコールの中にはMMHdlSizeのようにハンドルのサイズを求めるものがあります。ハンドルやポインタの中には，ASCIIZ型文字列(ASCIIコード列＋終端文字を表す0）しか代入しないということにすれば，strlen()関数などを用いて調べられそうなものですが，ポインタにはウィンドウレコード，ハンドルにはコントロールコード，テキストレコードなど，さまざまな種類のデータが代入されます。

実際のところ，ポインタやハンドルの大きさを示す情報はデータ領域本体とは別なところに存在します。その情報を管理するのがメモリマン（メモリマネージャ）です。ということは，メモリマンの関知しないポインタやハンドルはメモリマンを通じて大きさを収得できないということです（ということは，大きさを変更するということもできないことになるわけです）。

つまり，SX-WINDOWのプログラムを作成する場合，メモリポインタといったときは（メモリマンを経由して収得された）メモリポインタを指し，ハンドルといったときは（メモリマンを経由して収得された）ハンドルを指すわけです。

そんなことをいっても，

```
printf("Hello World !!¥n");
```

とすると，Cコンパイラはメモリマンとは無関係にポインタを作成してしまいます（この場合，printf()関数に渡されるのは，"Hello World !!¥n"という文字列そのものではなく，文字列が収められているデータ領域のアドレスです)。

で，本来ならば，このようなアドレスを指し示すデータを「ポインタ」と呼び，前述のものを「メモリマン経由で収得されたポインタ」と呼ぶべきなのですが，SX-WINDOWのプログラミングを行う際，前者のほうが後者よりも圧倒的に使用頻度が高いため，

ポインタ　→　疑似ポインタ

メモリマンを経由して収得されたポインタ　→　ポインタ

と呼ぶようにします。

つまり，SXコール一覧表などで「疑似ポインタが使用できます」「疑似ハンドルが使用できます」とかかれていた場合には，そのコールはメモリマン経由でそのポインタやハンドルの大きさを調べたり，変更したりしないということです。

今回使用したTSSearchも，疑似ポインタが使用できるコールのひとつですが，メモリマン経由でポインタの大きさを調べたりしない代わりに，あらかじめ，プログラマが90バイト以上の領域を確保しておく，という約束ごとがあります（なんで「インサイドSX」には載っていないのだろう？)。

きたわけですが，実用段階までは，いま一歩という感じです。迷子になってしまったファイルを探すには，そのファイル名を完全に覚えている必要があります。どこにやってしまったかすら記憶にないのに，ファイル名を完全に覚えている，というのは希な話ではないでしょうか。

あやふやな記憶から，ファイル名を類推し，その記憶場所を表示してくれるようになれば，より便利なプログラムとなってくれるはずです。

あやふやな記憶から具体的な名前を推定するという機能を完全に実現するというのはたいへんな処理ですから，ここではワイルドカードを用いて指定されたファイル名に合致するファイルをすべてのドライブから探し出すというプログラムにします。

TSSearch()関数でもワイルドカードを含むファイル名を検索できましたが，合致するファイルが複数あった場合，いちばん最初に見つけたファイルを拾って終了してしまいます。

ar.x
as.x
attrib.x

というファイルがあった場合，「a*.x」という条件で検索を行うと，見つけることができるのは，最初の「ar.x」のみです。ですから，TSSearchFileとは別にワイルドカードを含むファイル名に合致するファイルを探しまくる関数が必要です（リスト3）。

いろいろな機能を持った関数のようですが，具体的には，

1) mode2＝1でさまざまな初期化
2) mode2＝2で検索。この場合，条件にあうファイルが見つかるか(戻り値が正)，ひとつのドライブの検索が終わるか（戻り値が－8519)，どちらかだった場合，2)を繰り返す。それ以外の場合は検索終了かファイルエラーなので3)へ進む
3) mode2＝3でさまざまな終了処理

となります。

これをプログラムにすると，リスト2のようになります。

このプログラムは，ハードディスク，フロッピーディスクなどを問わず，すべてのデバイスに対し検索を行います。

実行してみるとわかると思いますが，リスト2では，検索結果を表示する場所(Text2)が2行分しか取ってありませんので，表示できるファイル名も2つまでとなっています。ウィンドウサイズを大きくして，表示領域も大きくすれば，大きくした分だけ，ファイルも多く表示できるようになります（ウィンドウデザイナでリスト2を読み込んでマウスで楽々操作，ということになっております)。

たとえば，使い込まれた大容量のハードディスクに対し「*.x」というファイルを検索した場合，条件に合致するファイルは10や20ではありません。当然，画面に収ま

リスト1

```
▼Window Size (269,133),0,0,0,ファイル検索
▼1,Text1 (24,24,252,52),0,0,0,0,3,1,0,1,
func Text1_KeyIn(text;str)
  TSSearch(text)
endfunc
▼1,Text2 (24,84,252,116),0,0,0,0,3,0,0,1,・
func Text2_Click()
endfunc
▼1,Text3 (24,60,256,80),0,0,1,1,3,0,1,0,↓
func Text3_Click()
endfunc
▼1,Text4 (24,4,252,24),0,0,0,1,3,0,0,0,検索するファイル名 (ワイルドカード可)
func Text4_Click()
endfunc
func TSSearch(filename;str)
  int i,ret
  int name = A_line(&ha01e, 90;l)
  int path = A_line(&ha01e, 90;l)
  str getname[90]
  for i = 0 to strlen(filename)
    poke(name + i, asc(mid$(filename, i + 1, 1)))
  next
  poke(path, 0)
  ret = A_line(&ha403, path;l, name;l, name;l)
  getname = ""
  if(ret < 0) then {
    alart(1, filename+"は見つかりません")
  } else {
    for i=0 to 90
      if(peek(name + i) = 0) then break
      getname = getname + chr$(peek(name + i))
    next
  }
  A_line(&ha02f, name;l)          /* MMPtrDispose
  A_line(&ha02f, path;l)          /* MMPtrDispose
  Text2.caption = getname
endfunc
```

リスト2

```
▼Window Size (269,133),0,0,0,ファイル検索
▼1,Text1 (24,24,252,52),0,0,0,0,3,1,0,1,
func Text1_KeyIn(text;str)
  TSSearch2(text)
endfunc
▼1,Text2 (24,84,252,116),0,0,0,0,3,0,0,1,
func Text2_Click()
endfunc
▼1,Text3 (24,60,256,80),0,0,1,1,3,0,1,0,↓
func Text3_Click()
endfunc
▼1,Text4 (24,4,252,24),0,0,0,1,3,0,0,0,検索するファイル名 (ワイルドカード可)
func Text4_Click()
endfunc
func TSSearch2(filename;str)
  int i,ret
  int sname = A_line(&ha01e, 90;l)
  int dname = A_line(&ha01e, 90;l)
  int path  = A_line(&ha01e, 90;l)
  int sfile = A_line(&ha01e, 268;l)
  str getname[90]
  di()                        /* タスクを占有
  encross()                   /* カーソルを踏切にする
  for i = 0 to strlen(filename)
    poke(sname + i, asc(mid$(filename, i + 1, 1)))
  next
  poke(path, 0)
  ret = A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h101;w)
  while(ret > 0 or ret = -8159)
    ret = A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h102;w)
    if(ret > 0) then {
      getname = ""
      for i=0 to 90
        if(peek(dname + i) = 0) then break
        getname = getname + chr$(peek(dname + i))
      next
      print getname
      Text2.caption = Text2.caption + getname + chr$(13)+chr$(10)
    }
  endwhile
  A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h103;w)
  A_line(&ha02f, sname;l)        /* MMPtrDispose
  A_line(&ha02f, dname;l)        /* MMPtrDispose
  A_line(&ha02f, path;l)         /* MMPtrDispose
  A_line(&ha02f, sfile;l)        /* MMPtrDispose
  decross()                   /* カーソルを元に戻す
  ei()                        /* タスクの占有を解除
endfunc
```

るものではありません。いくら表示領域を大きくとっていったとしても，限りがあります。

SX-BASICからダイレクトコマンドで，

Text2.editable = 1

とするとText2が編集可能なテキストアイテムになります。表示領域を編集してもしょうがないのですが，カーソルが移動できるようになりますので，表示内容をスクロールしながら眺めることができるようになります。

プログラムの改良（その2）

リスト1，リスト2は指定されたファイル名を検索し，その「ドライブ名」＋「フルパス」＋「ファイル名」で示すというものでした。入力も文字列情報ならば，出力も文字列情報という，まさにコマンドシェル上のwhere.xに相当するプログラムです。

where.xで「＊.x」形式のファイルをすべて検索しようとすると，出力が複数ページにわたり使いづらかったのですが，今回のプログラムを使えば，テキストのスクロールが可能ですので，そういう点では便利になっていると考えられなくもありません。

せっかくのSX-WINDOW環境ですので，もう少し別なアプローチを考えましょう。

入力を文字列で与えるのはともかくとして，ファイルの場所を文字列で示すというのはどこかしら不自然です。

ファイルを検索したあと，やることといえばファイルの大きさを調べたり，シャーペンで内容を表示したり，あるいは，SX-BASICで実行させたり，……といった動作が続くわけです。これらの動作は，そのファイルアイコン上でマウスのダブルクリックを行うことによって，実行されるようになっていますし，ファイルの移動/削除，名前の変更などは，ファイルアイコンをマウスでクリックしたりドラッグしたりすることによって行います。

ですから，ファイルの検索結果は文字列で表示するよりも，そのファイルのファイルアイコンがデスクトップ上に現れたほうがより便利でしょう。

というわけで，リスト2のうち，ファイルの出力結果を示すText2と，「↓」を表示するためのアイテムText3が不要になります。アイテムが減った分，ウィンドウの大きさも小さくしておきましょう((269,133)→(269,64))。

ファイルを検索する部分は，リスト2のままでよいのですが，その結果をテキストアイテムではなく，ファイルアイコンで示すように改造します。ファイルアイコンを示すプログラムは，「di.r」です（囲み参照）。

ドライブ「A」のディレクトリ「bin」内の「file.opm」というファイルを表示させるには，

fock("di.r a:¥bin¥file.opm")

とします。

ですから，リスト2の，

Text2.caption=Text2.caption＋getname

というところを，

fock("di.r " ＋ getname)

に置き換えれば，とりあえずは動作します。ですが，これですべての問題が解決するわけではありません。

「＊.x」というファイルを検索したとすると，

a:¥bin¥ar.x
a:¥bin¥as.x
a:¥bin¥attrib.x
⋮

という結果が返ってきたとします。

まず，いちばん初めの「ar.x」を発見した時点でディレクトリ「a:¥bin」の内容を示すウィンドウが開かれます。このとき，「ar.x」のファイルアイコンのみが選択状態で表示されます。

次に，「as.x」でディレクトリ「a:¥bin」の内容を示すウィンドウを開くようにSX-

リスト3

```
▼Window Size (269,64),0,0,0,ファイル検索
/* 最後に開けたディレクトリの名前
str     lastdir[90]
▼1,Text1 (24,24,252,52),0,0,0,0,3,1,0,1,
func Text1_KeyIn(text;str)
  TSSearch2(text)
endfunc
▼1,Text4 (24,4,252,24),0,0,0,1,3,0,0,0,検索するファイル名 (ワイルドカード可)
func Text4_Click()
endfunc
func TSSearch2(filename;str)
  int i,ret
  int sname = A_line(&ha01e, 90;l)
  int dname = A_line(&ha01e, 90;l)
  int path  = A_line(&ha01e, 90;l)
  int sfile = A_line(&ha01e, 268;l)
  str getname[90]
/* di()を有効にすると、検索がすべて終わってから、
/* ディレクトリを表示するようになります。
/*di()                  /* タスクを占有
  encross()             /* カーソルを踏切にする
  for i = 0 to strlen(filename)
    poke(sname + i, asc(mid$(filename, i + 1, 1)))
  next
  poke(path, 0)
  ret = A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h101;w)
  while(ret > 0 or ret = -8159)
    ret = A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h102;w)
    if(ret > 0) then {
      getname = ""
      for i=0 to 90
        if(peek(dname + i) = 0) then break
        getname = getname + chr$(peek(dname + i))
      next
      OpenDir(getname)
    }
  endwhile
  A_line(&ha402, path;l, dname;l, sname;l, sfile;l, &h103;w)
  A_line(&ha02f, sname;l)      /* MMPtrDispose
  A_line(&ha02f, dname;l)      /* MMPtrDispose
  A_line(&ha02f, path;l)       /* MMPtrDispose
  A_line(&ha02f, sfile;l)      /* MMPtrDispose
  decross()                 /* カーソルを元に戻す
/*ei()                      /* タスクの占有を解除
endfunc

func OpenDir(name;str)
  int   i
  str   path[90]
  i = strlen(name)
  path = name
  /*ファイル名を削除
  while(i > 0)
    if(path[i] == '¥') then {
        path = left$(path, i+1)
        break
    }
    i = i - 1
  endwhile
  if(name <> lastdir) then {
    lastdir = name
    fock("di.r "+ path + Text1.caption)
    encross()              /* di.rがマウスカーソルを戻してしまうので
  }
endfunc
```

BASICは命令しますが，「di.r」は，そのウィンドウはすでに開かれていると判断し，この命令を無視します。このとき，選択状態のファイルアイコンは，「ar.x」のみで，「as.x」は無視されています。

続く，「attrib.x」のときも無視され，結局，ディレクトリ「a:¥bin」の中で，選択状態にあるのは，「ar.x」のみ，という結果になってしまいます。

検索条件に合致するファイルアイコンすべてが選択状態にあるべきです。

この場合の問題は，「di.r」に検索条件を正しく伝えていないというところにあります。

つまり，個別のファイル名ではなく，

fock("di.r a:¥bin¥＊.x")

というワイルドカードを含んだ指定を行えば，この問題は解決されるのです（ドライブ名，パス名にはワイルドカードが使えません）。

TSSearchFile2によって検索されたファイル名には，

a:¥usr¥src¥push_bon.bas

のように必ず「¥」が含まれています。ですから，文字列を右から見ていき，初めて「¥」を見つけたところで文字列を左右に切断すると，

右側：（個別の）ファイル名

左側：そのファイルの存在するディレクトリの名前

ということになります。ですから，

左側の文字列 ＋ Text1.caption

で，「di.r」に渡すべき引数を得ることができます。

ついでに，前回開いたディレクトリと同じものを開くよう命令するのは，明らかに無駄なことですから（「di.r」に無視されるのは，わかりきったことですので），ディレクトリ名を得た時点で，その判断を行い，不要な命令を行うことを避けています。

「di.r」は，ファイルアイコンの表示中，マウスカーソルを踏切に変更しますが，表示が終了すると，以前の状態に関係なく矢印型に戻してしまいます。ひとつのディレクトリを表示したからといって，ファイル検索がすべて終了するわけではありません。勝手にマウスカーソルを矢印に戻されても困ります。そこで，「di.r」を起動したあとに，もう一度SX-BASICのほうで，マウスカーソルを踏切型に戻してやる必要があります(63行。この一件は「di.r」やSX-BASICに原因があるのではなく，SX-WINDOWのイベントマンにあるものだと思われます)。

リスト2のように，ファイル検索中，他タスクの実行を禁止しておくと，検索が速くなる，検索中にほかのプログラムからファイル操作をされることがなくなるなどメリットがあるのですが，今回のプログラムでそれを行うと，ちょっとした不都合が起こります。

ファイル検索というのは，けっこう時間のかかる作業です。大容量のハードディスク上をくまなく探し回るわけですから，容量が大きくなればなるほど，検索に時間を要するわけです。また，理由はよくわかりませんが，MOの検索もけっこう時間がかかります。

ファイル検索中に，他タスクの実行を禁止しておくと，「di.r」の起動も禁止されます。禁止中に発令された命令は，一時ため込まれ，解禁後いっぺんに実行されます。

数分間の沈黙のあと，ファイルアイコンがどばっという状態よりは，多少検索時間がかかってもファイル検索中，逐次その経過が表示されるほうが望ましいと考え，あえて他タスクの実行は禁止しませんでした。

参考文献

1）吉沢　正敏，「追補版SX-WINDOWプログラミング」，ソフトバンク

di.r

SX-WINDOWのシステム本体はファイルの情報を表示するプログラムを持っていません。ディスクの内容を眺めたり，コピーを行ったり，という作業は専用のプログラムが起動され，システム本体とは少し別なところで行われています。そのような，SX-WINDOW環境を構成するプログラムには，何種類かあるのですが，そのなかでも，ディスクやディレクトリの内容を表示するプログラムが「di.r」です。

読者の皆さんのなかには，初めてこのプログラムの名前を聞く方がいるかもしれません。ハードディスク上を探し回ってもおそらく，見つけることはできないでしょう（SX用のものとしては）。

しかし，プログラムリストの掲載を忘れたり，入手先を掲載していないフリーウェアだったり，過去の付録ディスクに収録されたプログラムだったりということではありません。SX-WINDOWのシステムディスクにちゃんと収録されています。

SX-WINDOWのシステムディスクを検索すると，「builtin.lb」というファイルが見つかると思います。

この中に「di.r」が入っています。この「builtin.lb」の中には，ほかにも「chd.r」（「ドライブトレイ」に関する処理を担当します）や「pinfo.r」（「プロセス情報」に関する処理を行います）というプログラムがコードリソース形式という特別な方法でまとめられています。

これはその流儀に則った方法でそれぞれのプログラムをバラしてやらなければならないので，非SX環境のコマンドシェル上からは実行できません。

新しいバージョンに変更する必要が生じた場合，あるいは，自分で作成したオリジナルバージョンのものと交換したい場合などはリソースエディタを用います。

このように，若干，収録のされ方が異なる「di.r」ですが，SX-WINDOW環境では先ほどの「builtin.lb」が自動的にばらされ，ほかのコマンドと同様に扱うことができるようになります。

扱い方は，コマンドラインから，

ls.r＋ディレクトリ名

で指令されたディレクトリを表示します。このとき，

ls.r＋ディレクトリ名＋ファイル名

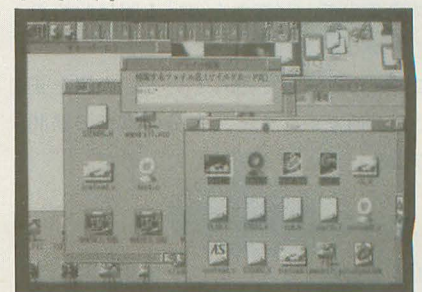
という指定をすると，そのファイルの存在するディレクトリを表示したあと，該当するファイルのファイルアイコンを反転表示します（反転表示されたファイルアイコンは，選択状態にあることを示します）。

コマンドラインの引数といっても，SX-WINDOWを使っている限りあまり使わないものですが，SX-BASIC上からはfock()関数を用いて使用することができます。

例）

fock("di.r a:¥usr¥src¥＊.sxb")

fock("pinfo")

また，OPT.1キーを押しながらクリーナーアイコンをダブルクリックすると写真のようなコマンド入力用のウィンドウが開かれますので，ここのテキストを編集し，命令を入力することもできます。

写真：コマンド入力用ウィンドウ

DOS/V退院する

Takazawa Kyoko
高沢 恭子

おつきあいの距離のとりかたもいろいろですが，仲良くなるまでの距離や問題も，またそれぞれ。ひとも，猫も，パソコンだって，あせらずに丁寧に時間をかけてなじんでいくのがよいでしょうか。

はじめてたずねる家をさがしながら路地裏などを歩いていると，どこからか小さなこどもがあらわれることがある。

その子はこちらを気にかけながら，前方へすこしいっては立ち止まり，ふりむいてはまたすすむ。つい誘われるように歩いているうち，けっきょくたずね当てる家の前まできて，その子はそこでピョンピョンはねたりする。

案内してくれたのだろうか。でも私がさがしていた家がなぜわかったのだろう。なんとなくお礼みたいな気持ちでニコッとすると，はにかんだようにユビをくわえたり，体をくねらせたりする。

その家がふだんから来客の多い家で，こどもたちもそれをよく知っているのか。そうではなくて，その子は見知らぬ私がどの家をおとずれるのか，偵察しながらいっしょに歩いていただけなのか。

たいていのこどもはお客さんが好きだ。めずらしいし，ニコニコしていて，おみやげもある。両親もお客さんに合わせてニコニコするので，お客さんのいる間は叱られることがない。

こどものころ外で遊んでいて，衣服を正した人が通りかかると，自分の家への訪問客ではないかとよく期待したものだ。

山頂の野菊

山道を案内するネコの話を新聞で読んだことがあったが，それが京都の笠置山だったとは知らずにドライブした。

でも残念なことに，その案内猫「笠やん」はことしの2月に死んでしまっていた。3年くらい前，全国に名前が知れわたってから，観光客の過保護やイタズラなどですっかり弱ってしまったためだそうだ。

夫はいつも，週末のドライブコースを会社の人たちに推薦してもらう。古くから三重県内に暮らし，ドライバーとしてもベテランという人たちは，歴史の宝庫である近畿の各地について知りつくしているので，毎回いくつもの候補をあげてくれる。

そんなとき，あそこだけはやめたほうがいいですよという場所も，いっしょにいくつかあげられる。それはたとえば，落石が多い所であるとか，対向車がきたら万事休すというような狭い道であるなど，運転上の困難を旨としたものだ。

ドライバー2年生の夫としては，そうした悪路，難路とされたところも，運転の研修の参考にさせてもらっている。これまでも，大きな落石がゴロゴロしている崖ふちの山道，「タヌキに注意」の無照明の夜道など，いくつも走ってみた。

元弘の乱(1331年)で，後醍醐天皇が鎌倉幕府打倒をもくろんで籠城したという京都の笠置山については，とくに誰からの推薦も注意もなかった。歴史の匂いがするドライブコースとして夫が選んだものだったが，登山道の道路状態はどうやら運転上級者向きだったようだ。

5分ほどの道のりは，舗装されてはいるが，急な勾配や細い道幅の連続。対向車に出会うたびに，くずれ落ちそうな路肩いっぱいのすれちがいをくりかえし，それでもなんとか山頂についた。

野の花と缶詰がそえられた小さな墓標を山頂の「行場めぐり」コースの入口で見つけた。「笠やんの墓」という文字が読めたので，山にゆかりのある人のお墓かとさらによく見ると，その上に「案内猫」と書いてある。もしやここが，新聞で話題になったネコがいた山だったろうか。

社務所にいた住職夫人にたずねたところ，小さなお墓の主はまちがいなく新聞で見たあの案内猫だった。おみやげといっしょに並べられていた小さな冊子，コピーライターの森義久さんが著した詩と写真の本「笠やんの詩」をもとめて帰宅した。

マンションにもどって朝日新聞のデータベースから，「笠置山 AND ネコ」をキーワードに関連記事を検索した。

昨年の掲載が2件，ことしが1件で，まとめるとつぎのようなものだった。

体長約40センチ，白と茶のトラジマのオス猫は，笠置山に住みついて「笠やん」と呼ばれていた。あるとき山頂周辺の「行場めぐり」コース(約1km)を観光客について回ったところエサをもらい，それから人々を先導するようになった。

このことがネコの専門誌に紹介されて一躍有名になり，また笠置山がNHKの大河ドラマの舞台になったこともあって，観光客が以前の5倍近い，1日平均150人になった。観光客がカマボコやキャットフードを与え，笠やんは太りだし，食べすぎて吐くほどになった。また笠やんをとり囲んでほうり投げて遊ぶ客もいた。

笠やんは体調をくずし，観光客の悪ふざけをきらって1年くらい前からはしだいに姿を見せなくなっていたが，ことしの2月2日，息をひきとった。人間なら還暦に近い年齢で，笠置山には5年ほど住んだことになるそうだ。

写真集で見る笠やんは，まん丸顔の利口そうなネコだ。きっと案内のつもりではなく，お客さんを好きな人間のこどものように，よそからくる人がめずらしくて，いっしょに歩いていただけなのだろう。「全国的な人気者になったばかりに，人間の身勝手さにふり回されてかわいそう」という住職夫人の言葉も掲載されていた。

気むずかしい頭脳

どうしてもプリンタに印刷させることができなかったDOS/Vマシンが，Y氏のもとから退院してきた。じつはほかにも，一部の動作がおそいなど気がかりな点もあったのだが，どうやら原因と思われることがわかり，CPUの交換までにはいたらなかったという。

マシンの入院中に購入しておいた描画ソフト「SuperKID」を使ってみるのも楽しみだった。さっそくインストールして，テスト版のラフなスケッチを描き，これをプリントしてみた。

OK！　DOS/Vマシンでのはじめてのプリントができた。

ただし「SuperKID」での「ページ設定」に無頓着だったため，「ドキュメントサイズ」で印刷されたので，びっくりするほど小さな絵ができあがった。

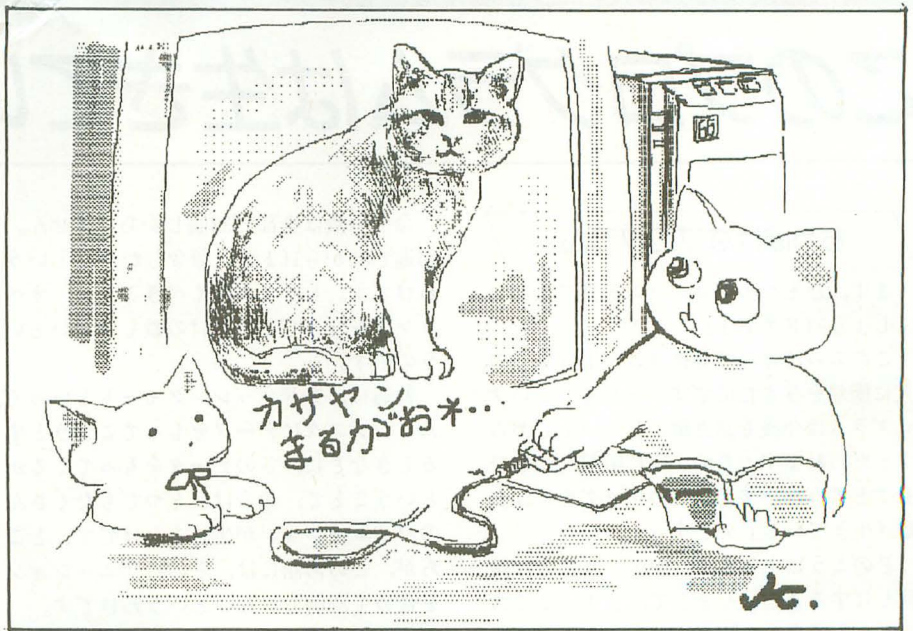
ドキュメントサイズと画面上で描いているときの便宜的なサイズ，また印刷時に設定されるサイズは，それぞれわけて理解していないと失敗する。さらに用紙内での配置の考慮もあるから，慣れるまでにはたくさんのミスがあることだろう。

ところでわが家のDOS/Vマシンが，なぜプリントの命令でつまずいてしまったのだろうか。

DOS/Vマシンはソフトがインストールされるたびに，周囲の状況を把握しながら，必要に応じて自動的に内部での書き換えが行われるのだそうだ。

とくにわが家のマシンは，基本ソフトの設定の折に，テストもかねてあまりにもいろいろなソフトがインストールされた。そのために内部の書き換えが複雑になりすぎてしまった。あらたにプリンタドライバを組み込んだあたりで，書き換えの項目が互いに影響しあって，なんらかの障害が出たものではないか。

そこでY氏はハードディスク内のすべてを消去して，設定のしなおしをしてみたのだそうだ。病因はともかく，治療の効果はあった。わが家のプリンタは動くようになったし，ほかの動きについても問題がなくなったようだ。

DOS/Vマシンが便利でパワフルであることは，すこし使ってみれば誰でもわかる。ただし，なかなか気むずかしいマシンでもある。また，むやみにソフトをふやすのは禁物で，メモリが激減する。いらないものは消してくださいよと，Y氏も帰りぎわにいい残していった。

illustration：Kyoko Takazawa

ソフトとの長い時間

「SuperKID」はまだ触りはじめたばかりだから，不慣れでわからないこともたくさんある。

マシンもソフトも，長い時間をいっしょにすごさないと，身につかない。評価もできない。そんなあたりまえのことが，このごろやっとわかってきた。

新しいソフトに接するときには，心がまえがいる。これからこのソフトのためにたくさんの時間をかけよう。偏見をすててファイトを持とう。できるだけマシンの前にいる時間をふやそう。それがいちばんモノをいう。

7年くらい前に，描画のソフトを見たときの印象は，10年前にワープロを見たときとよく似ていた。

手で文字の書ける自分が，なぜ機械を使って無表情の活字を並べなければならないのか。疑問と興味を両方感じながら使ううちに，ワープロも進化した。それとともに，自分の暮らしのなかで，手書き文字とワープロ文字の活動領域のちがいが，おのずからハッキリしてきた。プライベートの手紙のようにオリジナルを強調したい場合と，書類や原稿などのように大量生産やデータ化を必要とする場合とに，大きくわけられた。いまではもうワープロのない生活は考えられなくなった。

描画のソフトを見たときも考えた。絵を勉強したつもりの自分が，なぜこんなことをするのか。しかし，1年もしないうちに，その分野のちがいを認めてしまった。これは表現の技術も方法もちがう，新しいジャンルの絵だ。

利点も大きかった。準備がいらない。用紙と用具が無尽蔵。衣類もまわりも汚れない。絵の具や油の臭気もない。しくじったときに以前の状態に完全に戻れる。

利点とひきかえに，スケールが小さいとか筆勢やエネルギーがないなど，欠けている要素もあげればキリがないが，時間的なトクがなんといっても大きい。

そんなソフトの便利さに頼ってばかりいる私に，ちょっとした刺激がおとずれた。

「童話の挿絵をぜひアナタに描いてほしいってたのまれたんだけど……」

新宿のおばあちゃんが電話してきたのは，数カ月前だった。

教員時代の同僚だった男性が1年前に亡くなられた。以前から童話や詩の著作が多く，いくつもの賞を受けた人なのだそうだ。長編の作品が原稿のまま残されているので，夫人が本にしたいと望んでおられ，その挿絵を私にと希望されているということだった。

(以下次号)

第87回　知能機械概論——お茶目な計算機たち——

このプログラムは生きているのか!?

ちっぽけなプログラム

まず，ひとつのプログラムを見てもらいましょう（リスト1）。

このちっぽけなプログラムが実は多くの人に衝撃を与えたのです。そして，このプログラムは今後も語り継がれていくであろうと思われます。どのように衝撃的かをひとことでいいますと，このプログラムは実は「生きている」のですっ！

どのように生きているかということは後回しにすることにしまして，まずはこのプログラムの内容を細かく見ていくことにしましょう。

このプログラムはニーモニックで表記されています。つまりアセンブリ言語で書かれているのです。アセンブリ言語とはなにかといえば，要するに，C言語，Pascal言語などとは違い，計算機が直接実行できる機械語命令（0と1の列）にそのまま変換することができる言語のことです。

ちなみに，このプログラムを全部機械語に直すと，リスト2のような80個の命令(16進数表記してます）の列になります。

ただし，計算機が直接実行できるといっても，ここに並んでいる命令を直接実行できる計算機は実在していません。ですから，仮想的な機械語プログラムということになります。

言語の特徴

この仮想的な言語には2つの大きな特徴があります。

1．命令の少なさ

リスト1

```
1111
find            0000(start) ->     bx
find            0001(end)   ->     ax
calculate size              ->     cx

1101
allocate daughter           ->     ax
call            0011
cell division
jump            0010

1100
save registers to stack
1010
move            |bx|    ->  |ax|
decrement       cx
if cx == 0      jump 0100
increment       ax & bx
jump            0101
1011
restore registers
return
1110
```

命令の数は全部で32個しかありません。16進で00から1fまでの命令しかないというわけです。しかも，驚くべきことに，オペランドも含んでそれだけの数しかないというのです。

普通は，アドレッシングモードといって，あるレジスタにデータをもってこようとするときなどに，どのデータをもってくるかということで，命令はひとつでもたくさんのバリエーションが生じるわけです。ところが，この言語には，そのバリエーションを含めて32個しかないというわけです。

整数を直接レジスタに入れることはできません。たとえば，4という数字をcxレジスタに入れるには，

zero or1 shl shl shl

という5つの命令を必要とします。

まずcxレジスタの中身をゼロにして，次にそれ（最下位ビット）を反転して1にし，2回だけ左シフトすればやっと4になるのです。

2．テンプレートアドレッシング

たとえば，JUMP命令を実行する場合，アドレスを直接指定したり，相対的な位置を指定するなどして，目的のアドレスに飛ぶのが普通ですが，これはそうではなく定めたパターンに一致したところにジャンプします。

JUMP NOP_0 NOP_0 NOP_0 NOP_1

というように5命令が並んでいた場合，後ろの4命令の数字をひっくり返した，

NOP_1 NOP_1 NOP_1 NOP_0

という命令の並びを探し，その後ろのアドレスにジャンプします。

この機能はこのシステムのツボです。JUMP命令があると，その後ろに並んでいるNOP命令（NOP_0かNOP_1の組み合わせ）パターンを補完するようなパターンを現在の位置から前方と後方に検索し，近いほうにジャンプする（なければエラーフラグをセットしてそのまま次の命令を続ける）のです。

リスト2

```
01 01 01 01 04 02 03 03 18 1c
00 00 00 00 07 19 1d 00 00 00
01 08 06 01 01 00 01 1e 16 00
00 01 01 1f 14 00 00 01 00 05
01 01 00 00 0c 0d 0e 01 00 01
00 1a 0a 05 14 00 01 00 00 08
09 14 00 01 00 01 05 01 00 01
01 12 11 10 17 01 01 01 00 05
```

ところで，このNOP_0とかNOP_1というのはこの場合にオペランドの意味しかもちませんが，れっきとした命令です。JUMPのような命令が前になければ，No Operationという命令（なにもしないという命令です）になります。

この2つの特徴にはかなり強い意図が込められています。命令数を少なくしたことは遺伝子コードの数を意識したものですし，テンプレートマッチングというのは，タンパク質が細胞内でお互いに相互作用する場合の方法，つまり，タンパク質の表面がお互いに補完するときに作用が起こるという現象を意識しているのです。

衝撃のプログラムの意味

このプログラムの意味を見ていくとしましょう。まず，先頭の1111と最後の1110はプログラムの最初と最後を示すマークです。1111といっても，実際は，

NOP_1 NOP_1 NOP_1 NOP_1

のような4命令のことですが，テンプレートマッチングの際にはマークとしての別の意味が生じてきます。

最初のブロックでは，自分自身の先頭と終わりを探しにいき，その場所(アドレス)をそれぞれbxレジスタとaxレジスタに入れます。そして，引き算をして，自分の大きさを知ります。

最初のfindの行は，具体的には，

adrb（address backward）　NOP_0 NOP_0 NOP_0 NOP_0

という5命令で構成されます。backwardを表すbがついていますので，アドレスをさかのぼって，

NOP_1 NOP_1 NOP_1 NOP_1

というパターンを探します。そして，そのパターンの最初のアドレスがbxに入ります。2つ目のfindの行は，同様に，

adrf（address forward）　NOP_0 NOP_0 NOP_0 NOP_1

で，今度はシッポを探します。

2つ目のブロックは自己複製をするところです。このプログラムのメインループで，ループを1回まわるごとに自分（親細胞）と同じプログラム(娘細胞)，つまり命令列そのものを別のところに複製し，その複製

部分の実行を開始させます。

allocate daughterは，娘細胞のためにオペレーティングシステムからメモリを割り当ててもらい，その先頭アドレスをaxレジスタに入れます。そして次に，0011ですから1100というパターンから始まるブロック（コピーを行う部分）を呼び出します。

cell divisionとは，いわゆるプロセスのfork命令と同様で，娘細胞に命を吹き込むものです。具体的には，娘細胞内への書き込み特権を放棄し，娘細胞に実行開始アドレスを渡して，実行する待ち行列に加えるというものです。最後にまたこのブロックの先頭に戻ります（無限ループ）。

最後のブロックは実際にコピーを行う手続きです。スタック（サイズは10）へax,bx,cxレジスタの中身を待避したのち，親細胞の大きさだけ，その中身（命令）を娘細胞の場所に順番にコピーしていきます。

move　|bx|　->　|ax|

というのは，bxレジスタの内容をアドレスとしてもつメモリの中身を，axレジスタの内容をアドレスとしてもつメモリへコピーするということです。コピーが終わると最後のreturnでここを呼び出した命令の次に戻り実行を続けます。

なお，正確にいえば，リスト1には含まれていませんが，テンプレートの大きさ4をdxレジスタに冒頭でセットする部分と，プログラムの区切りとして1語分とっているので終わりのアドレスを＋1する部分が命令として必要になります。

実行環境

このプログラムとそれが実行される環境について，書くことにしましょう。

各プログラムの実行はひとつの仮想的な生物に対応します。最初は，このプログラムをひとつだけ実行開始します。プログラムの実行をプロセスと呼ぶことにします。ひとつのプロセスは，アドレスレジスタと数値レジスタを各2個，エラー条件を表すフラグジスタ，スタックポインタ，10ワードのスタック，命令ポインタを各1個もっています。

実行されたプロセス（ご先祖様）はすぐに娘細胞をcell division（divide命令）によって作り始めるので，どんどんプロセスは増加していきます。計算機に無数のプロセッサチップがあるのならば，それぞれに実行させればいいのですが，そうはいかないので，時間を区切って順番に少しずつ実行していきます（タイムスライシング）。疑似的に並列実行させるわけです。

実行する順番が回ってきたとき，何命令分実行するかということは係数をどのように設定するかによります。生物の大きさ（総命令数）によらない割り当てなら，生物の大きさに比例した命令数だけ実行されます。生物の大きさに応じて有利/不利に設定するのならば係数を変えて，割り当てサイクル数を増/減します。

ところで，メモリ領域（スープと呼びます）は60000バイト，つまり60000命令分という有限のものですから，少したつと，いっぱいになってしまいます。そこで，生物に死んでもらわねばなりません。寿命の概念です。メモリ領域で使用された割合が80%を超えるとreaper（刈り入れ人）が動き出して，生物を殺してメモリを回収し始めます。80%を超えていると基本的には生まれた順に殺していきますが，エラーを起こしたら順番は早まり，エラーなしで実行するのが難しい命令2種類をエラーなしで実行できたときには順番は遅くなります。

突然変異

テンプレートマッチングはどこにジャンプするかわからないという非決定性を生む重要な機能でしたが，さらに重要なのが突然変異です。これが予測もしないような生物を作り出す原因となります。

突然変異には次の3通りの起こり方があります。

1）メモリの中の命令をランダムに選び，ビットを反転（0なら1，1なら0）させる。これは，地球上における宇宙線による変異と同様な機能といえます。

2）コピーをする命令（mov_cd，mov_abなど）で一定の割合でコピーするデータのビットを反転させる。

3）低い確率で演算命令の結果をプラス/マイナス1させる。

要するにわざとプログラムの実行が最初に規定したとおりにはいかないようにするのです。ただしこの3つのタイプの突然変異のうち，2）と3）は生物集団が進化を経て，全体としてある程度複雑になったあとには，あまり意味のある影響はなかったということだそうです。

1）は予期しにくい生物が出現する原因となります。たとえば，NOP_1がNOP_0になったとき，それがテンプレートとして探される場合には，大変なことが起きます。マッチして実行されるはずだったことがされなかったり，とんでもないところ（隣の生物）の手続きが実行されることだって起きてしまいます。自分の先頭とシッポの確認さえ間違えてしまい，長さの異なった娘細胞を作ってしまったりもします。

いい忘れましたが，別の生物の領域に関して，書き込みはできませんが，読み出しや実行はできるのです。近くにいる生物の命令を実行できるというのは，他生物との相互作用を意味するのです。

何が起こるのか？

仮想的な機械語ですから，ひとつの命令ごとにそれを翻訳しながら実行するようなプログラムを作ればどんな計算機においても，このプログラムは実行できるのです。

さて，先祖様をひとつ，メモリの中に入れて実験を開始します，ちょうど，シャーレの培養液に注射器で細菌を注入するように。実際には，東芝製のラップトップパソコン（80386＋80387）の中で実験は繰り返されました（長期間の進化の様子を調べるときには，大型計算機を使います）。

実験を開始するとすぐにスープの中はいっぱいになってきます。たぶん数分のことでしょう。reaperが動き出します。だいたい1人だけ娘を作ると寿命になり死んでしまうというのが平均的なケースになります。スープ内は，いろいろな命令列をもち，いろいろな大きさの生物でバラエティに富んできます。

その中には注目すべき生物がいます。寄生生物です。それは自分自身にはコピー能力がないのです。つまりコピーするルーチンを含みません。しかし，コピーができる普通の生物が近くにいるとその機能を借りて（ほかの生物のコピーする手続きを一時的に実行する），もちろんコピー手続きの含まれない自分自身をコピーするのです。寄

このプログラムは生きているのか!?

生といっても，寄生された生物はメモリやプロセスの実行時間のぶんどり合戦が厳しくなるという意味だけで，直接には害はありません。

そのうち，そのような寄生に合わないタイプの生物が生まれます。つまり，免疫ができるのです。しかしその免疫を無効にするような新たな寄生生物が生まれます。さらに時間が経過すると，寄生する生物を逆に利用してしまう生物，つまり重寄生生物も生まれてきます。

さらには，ひとりではふつうに生きているが似たようなものが集まるとお互いに命令を利用し合うような現象（社会的寄生）も観察されました。

寄生虫の実態

それでは，先祖生物に寄生生物がどのようにカラむのか，そして寄生生物にどのように重寄生生物がカラむのかを，進化の結果自然に生まれたプログラムを実際に示して説明することとしましょう（図1）。

3つのプログラムを左から順に示します。左側の先祖生物のプログラムはすでに示したものと同じです。JUMP命令などで制御が飛ぶところは実線でその様子を示してあります。

さて，近所に真ん中に示すような寄生生物がいたとします。これは，先祖プログラムのコピーをする手続きの冒頭の1100が突然変異によって1110に変わったために，プログラムの終わりのマークということになり，それ以下がコピーされなかったという種類の生物です。

しかし，この生物はコピーする手続きがなくても，娘細胞を作れるのです。それは，call 0011のところで，近くのコピー手続きのところに飛んで，自分の複製を作ることができるからなのです。図中では，ほかの生物の命令を実行しにいく線は破線にしてあります。

重寄生生物は，そのような寄生生物を逆に絡めとってしまうことができます。寄生生物がもしこの重寄生生物のコピー手続きを利用しようとしてそこに飛ぶとします。突然変異の結果，重寄生生物のコピーの手続きはreturn命令が切れてなくなっており，直接0011というパターンに飛ぶ（自己複製ブロックの中のcall命令の次の0011にマッチ）ようになっています。

したがって，寄生生物は自分のプログラムにもう二度と戻ることができません。さらにこの重寄生生物は突然変異の結果，1回自己複製するたびにプログラムの最初に戻ってあらためて自分の最初と最後を確認するようになっています。したがってこの寄生生物は寿命が尽きるまで重寄生生物を複製し続けるのです。

いやー，なんということが，計算機の中で起こっているのでしょうか？　ほとんど絶句です。

若者がどんどん寄生されて……

今回紹介したシステムはTierra（スペイン語で地球の意味）といい，記述言語の名前をTierranといいます。こんな大したことをやってくれたのは，Thomas S. Rayです。Delaware大学にいましたが，現在は人工生命研究の世界的拠点のひとつとなっている京都のATRの研究所にいます。

このプログラム（あるいは文献1）によって，こういう人工生命の研究に引き込まれた若者，あるいはこれから引き込まれていく若者は，たくさんいることでしょう。実際そういう声をよく聞きます。

僕はそれだけのインパクトがあることだと思います。あんなに簡単な80命令のプログラム（半分以上がNOP命令！）なのに，ここまで複雑な現象（ほかにももっと興味深い現象もあるのですが紹介しきれません）が生じるというのです。でもまあ，受け止め方も人それぞれでしょうから，このへんでやめておきましょう。

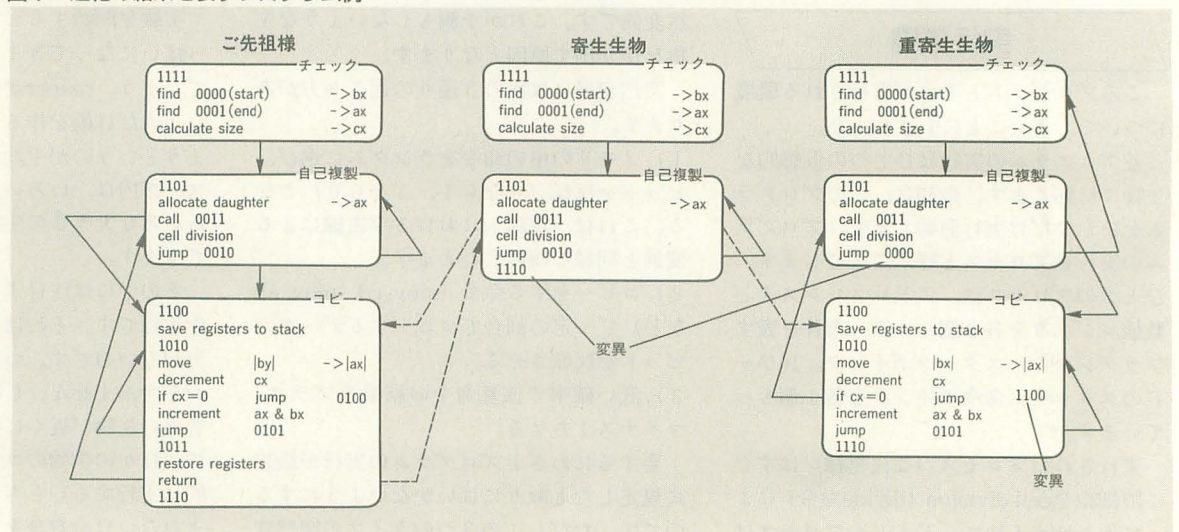
最後に，この仮想生物たちの奥さん（だと思いますが）のユーモアあふれることばを文献2から引用して今回は終わることにしましょう。

「彼らが実在しなくてよかったわ。彼らに食べ物をあげなくてはいけないし，家を埋め尽くしてしまうでしょうから」
──Isabel Ray

参考文献
1）T.S. Ray, "An Approach to the Synthesis of Life", Artificial Life II, pp. 371-408, 1991.（人工生命に関する2回目の国際会議の会議録）
2）T.S. Ray, "Evolutionary Approach to Synthetic Biology", Artificial Life, Vol. 1, No. 1/2, pp. 179-209, 1993.（創刊されたばかりの人工生命に関するジャーナル）

e-mailアドレス
ari@info.human.nagoya-u.ac.jp

図1　進化の結果を表すプログラム例

[第8回]

家庭用パソコン急加速

Ogikubo Kei
荻窪 圭

昔から，PDAにはCCDカメラをつけるべきだとか，1994年はビデオキャプチャとビデオ出力が流行るっていってきた(自画自賛モード)。私がこういうことをいうときはたいてい半分願望で半分本気なのだけれども，いやはや，本当に，そうなってしまった。

第1期はどうでもいいマシンが揃っただけだったんだよね。それこそ，X1にも劣る，みたいな。でも，パナソニックのWOODYが火をつけた。いろいろあるなかで，あれだけが「音質を重視」し，ビデオ画面を（静止画だけど）キャプチャする機能を持っていた。その分，CRTの質が悪いとか，まあ，いろいろとあったのだけど，評判がよかった。

で，秋，各メーカーが一斉に第2弾の製品を作り始めたわけだ。第2期っていうよりは，第1期part 2って程度だけど。

いちばんの傑作は，NECの文豪。パーソナルワープロにCCDカメラをくっつけてしまった。それだけではない，まずはイメージを定着させよう，ってことで「ビデオキャプチャ」って言葉を連呼するCFを作った。このあたりが凄い。このCFを何度も見れば，誰もが「ビデオキャプチャ」って言葉が頭にこびりつき，なんだか面白いことができそうだ，って気分になる。あとは，そのCFを見た人がビデオキャプチャ機能を持ったパソコンを探せばいいわけだ。このCFのおかげで，ビデオキャプチャ機能を持ったパソコンが身近になるわけで，NECさんありがとう状態なのだ。

それにしても，ワープロ専用機はやはり思い切ったことができていいね。NEC以外にもビデオ入力関係の機能を持ったワープロが出てきているみたいだし。あとは，キヤノンとかエプソンとかYHPと技術提携をして(OEMして)，フルカラーインクジェットプリンタを搭載すれば怖いモノなしだ。面白い。

ではパソコンはどうか。

◈第1期マルチメディアパソコンとは

マルチメディアパソコンって，何じゃらほい，っていっている間に，いつのまにか暗黙の了解的にマルチメディアパソコンの最低限のスペックができた。

曰く，サンプリング音源のサウンド機能を持っていること。

曰く，倍速CD-ROMドライブを内蔵していること。

曰く，256色以上の発色数のビデオカードを持っていること。

これだけだ。ここにどんな付加価値をつけるか。

ひとつはテレビが映ること。パーソナルユースとしてはテレビが映るってのは魅力的に映る。

ひとつはソフトウエア。Windowsったって，こういうパソコンに惹かれて買う人にとっては難解きわまりないもので，アップルがパフォーマシリーズで「At EASE」をつけたように，初心者向けのメニューソフトを内蔵して誰でも使えるようにしてしまえばいい。

ひとつは価格。いうまでもなし。

ひとつは独自機能。これらを踏まえて独自に何をつけてくるか。

このヘンをひっくるめると何が出てくるか。面白いのである。それがNECのPC-9821 CanBe(キャンビーって名前の是非はともかくとして）であり，コンパックのプレサリオCDS520（520って数字の是非はともかくとして）であり，アップルのパフォーマ630なのだ。いやはや，あっという間にここまで辿り着いたか，ってのが偽らざる気持ちだ。

どれも面白い。いままでの高性能低価格戦争から一歩離れて市場を広げようという意志がはっきりみえる。

製品レビューをするつもりはないが，ひととおり特徴を見ておこう。

まずはコンパック プレサリオCDS520である。

CPUは486SX2/66である。いままで家庭向けのパソコンは適当な性能でいいというメーカー側の常識から，この手のマシンにはSX/25かSX/33を載せてくるのが普通だったが，とりあえずSX2/66を載せてきたのは偉い。

モニター一体型でステレオスピーカー内蔵である。これはまあ普通。

メモリは標準で8Mバイトである。当たり前だが，いままでは4Mバイトしか搭載しないで売っていたのだ。拡張なしでひととおり動くこと，って条件をやっと満たした。

HDDは420Mバイトである。これも偉い。いままでは家庭用ってことで，HDDは250Mバイトクラスが常識だったからね。

サウンド機能も倍速CD-ROMドライブも当たり前である。コメントなし。

ディスプレイがVRAMを512Kバイトしか積んでない，800×600で256色しか出ないシステムなのだ。これだけがタコである。どうして，頑張ってVRAMを1Mバイトにしなかったのか。たったひとつの残念なところ。

FAXモデムを内蔵している。これも偉い。

付属ソフトも凄い。DOSとWindowsは別にして，前から搭載していたプログラムマネージャに代わるシェル「Tab Works」以外に，「オーガナイザー」「MS WORKS」「MaxFax」「PC PaintBrush」「A IV for Win」。CD-ROMとして「ピーターと狼」に「THE ANIMALS！」。至れり尽くせりである。付属ソフトの数ではアップルのパフォーマ630に負けるが，こんなもんだろう。

凄いのは価格である。249,000円なのだ。

続いて，NECのPC-9821Cb。例によって，Cb/Cx/Cfと3つ出ているが，いちばんパーソナルなのはCb。モニタ一体型で，足元にステレオスピーカーがある。

CPUは486SX/33である。これは残念。最低でもSX2/66が，できたらDX2/66が欲しかったところ。ちなみに，Cbも上位機種のCxもSX/33だが，最上位機種のCfだけはPentium/60で，この極端さが面白い。

メモリは標準で7.6Mバイトである。98Multiとしては偉い。

HDDは210Mバイトである。抑えめだ。たいして偉くない。

サウンド機能も倍速CD-ROMドライブも当たり前である。コメントなし。

ディスプレイは640×480でフルカラー。これはいい。高解像度モードがないけど，まあ，妥協できる線だ。Windows上でテレビを見るときは256色モードでないとダメだ，ってのがよくわからんけど，PC-98ってたいていどっかよくわからんところがあるから，まあ，いい。

FAXモデムはモデルによって違うが，いちばん安いヤツだとオプションである。これは×。

付属ソフトはDOSとWindowsは当たり前として，「MS WORKS」がある。これもいい。ほかにも簡易メニューなどがついている。が，動画キャプチャソフトが別売りなのは×。3,900円なら，バンドルすればいいのに(間に合わなかったのか？)。

テレビは当然見られる。テレビリモコンもついてくる。これはいい。

価格は325,000円。プレサリオに比べて高いが，実売価格を考えると，20万円台後半ってことで，許せる。

続いてパフォーマ630。

モニタは一体型でないがセットになっている。本体の上に載せてもたいした一体感などないデザインなので，どうせなら最初から一体型で作ってほしかったが，まあ，しゃあない。

CPUは68LC040の33MHz。特に問題なし。来年にはPowerPCになるだろう。本当はコプロつきがいいのだが。

メモリは8Mバイトである。少ないが使えないこともない。

HDDは250Mバイト。可も不可もなし。たいして偉くないが，SCSI標準装備なので増設は楽。

サウンドも倍速CD-ROMドライブも当たり前だが，付属15インチモニタのスピーカーは音が悪いので×。

ディスプレイは640×480で32,000色だ。Macintoshの標準的スペック。824×624で256色もあり。これはうれしい。

FAXモデムはオプション。これは×。

付属ソフトは化け物みたいに多い。アメリカのノリだ。コンパックに近い。「クラリスワークス」はともかく，ビデオ編集ソフトからハガキサイズに限定したワープロ，TrueTypeフォント，各種CD-ROMソフトまで。こいつはお得だ。

テレビリモコンもビデオキャプチャ機能も当然ついている。AV関係ではもっとも充実しているといえる。

価格はオープンだが，多分，20万円台後半で売られるだろう。

WOODYも新製品を出してきた。ヨーロッパブランドのオリベッティやフィリップスの動きも見逃せない。

可哀想なのはアイワだ。第1期に遅れてとんでもないキッチュなデザインのマシンを出してきたはいいが，サウンドもテレビも質はいいものの，ビデオキャプチャ機能やWindows上でテレビを見る機能がない。ただくっつけただけ。可哀想である。もう死んでいるのだ。

どうして死んでいるか。テレビとCDプレイヤーとしての機能を重視した画質のよいマシンではないか。せっかくのトリニトロン管ではないか。

アイワはパソコンよりもテレビを重視したモデルを出したのだ。パソコンとしても使えるテレビなのである。しかし，だ。テレビを中心にした人はテレビを買うのである。テレビの付加価値としてパソコンをつけると，テレビのみに対して20万円以上高くなるが，パソコンの付加価値としてテレビをつけると，ほんの僅かな価格アップで済むのだ。だったら，テレビを見ようとしてパソコンを買う人間なんているわけがない。

アイワ的なコンセプトが通用するとすれば第2期である。第2期パーソナルマルチメディアパソコン時代だ。

脱パソコンを目指しているのである

さて，今回の一連の流れはひと昔前の家庭向けパソコンとはどこか違ってきている。このへんを感じとれるのではないかとスペックを並べてみたわけだ。

まず，基本スペックであるが，「その時点で使いモノになる最低限のレベルはクリアしている」のである。メモリもハードディスクも同様だ。PCならば最低でも486SX/33（本当はSX2/66を最低ランクにしたいところだが），Macintoshならば68LC040/33。これが最低ランク。メモリは8Mバイト。HDDは200Mバイト。とりあえず，そのまんまで使いモノになる。

売りになりそうなオプションはすべて内蔵している。マルチメディア仕様，FAXモデム，テレビなどなどだ。とりあえず，内蔵できるものはみな内蔵した。おかげで，増設にまつわるトラブルは解消できる。これ以上何が必要というのか。

いちばんのポイントは，WindowsやMac OSという

OSをベースに，使いやすいシェルをかぶせ，取っつきをよくしたことだ。つまり「不要なファイルを見せない」ことである。ワンクリックでアプリケーションが立ち上がり，テレビも見られるし，CDも聴ける。誰でもワープロ専用機並みに簡単に使えるってことである。

そういう意味では，Macintosh化がどんどん進んでいるっていえる。Macintoshはサードパーティが苦労して作ってきた市場を，どんどん本体に取り込んで標準装備してきた。サードパーティにとってのうまみはどんどんなくなるけど，ユーザーから見れば，最初から内蔵されていたほうがトラブルもないし，使いやすい。ビデオカードしかり，CD-ROMドライブしかり。今度はビデオキャプチャカードときた。サードパーティの成長と頑張りでもってここまできた98やPCの世界でも，パーソナルユースのマシンはMacintoshのようになってきた。何でも本体に内蔵し，サードパーティはより独自性のある製品を安く作らねばならないという大変な状況になったということだ。バンドルしてもらえる製品はいいが，そうでないブランドはどんどんマイナーになっていく。

で，しかも安い。本格的に脱パソコンをし，家庭へそのまま入り込める下地を作れるように見える。

こうなってくると，各社の工夫はユーザーインタフェイスとなる。最初の数カ月使ったら飽きちゃうだろうなという付属ソフトをただ立ち上げるだけのアホらしいものから，プログラムマネージャの代替になるようなできのよいものまで。パソコンらしさを廃して馴染みやすくしたものから，あくまでもWindowsにいることを意識させるものまで。このあたりのバランスが難しい。

見ためが優しいのはいいが，いざ新しくソフトをインストールしようと思うとWindowsの世界に揉まれてわけわかんなくなるのも困るし，かといって，Windowsの呪縛が最初から見えていてとっつきにくいのも困る。

WOODYやプレサリオはあくまでもパソコンらしさを失わない設計になっているし，アイワやNECは全画面をオリジナルメニューで覆ってしまった。

どちらがいいか。難しいところだが，WOODYやプレサリオの手法がいちばんバランスが取れている気はする。あまりオモチャみたいなデザインになっても反発を招くのではないだろうか，ってことだ。買う人はまがりなりにも「パソコン」を求めているのだから。

第2期にはナニがどうなるか

いちばんのポイントはここだ。

古くからのパソコンユーザーは今回の流れにけっこう反発するだろう。最初から市販アプリケーションをバンドルするってことは，ユーザーのソフト選択の自由度を失わせることになるし，増設の面白さもそこにはない。ただ，買ってきて遊んで，その気になった人だけが，新たにソフトを買い揃え，うまくインストールできなくて困る，って構図は目に見えているからだ。

しかし，多くの人は，「選択の自由度」なんて与えられ

illustration : Haruhisa Yamada

たって困るだけなのである。これが真実だ。「選択」するってことは，何を選択できるのか知らねばならないし，1つひとつの選択肢がどのような特徴を持っているのか知らねばならない。そもそもパソコンに対して漠然としたイメージしか持っていない人たち，会社で使っている無味乾燥なDOSソフトのイメージしか持ってない人たちに，一から知識を吸収して覚えなさい，ってのはあまりに酷な話だ。であれば，そこそこ使えるソフトを最初からバンドルしていくのが親切というもの。だから，これらのアプローチは正しいのである。パソコンが，道具から，ワープロや住所録にもなる「メディアプレイヤー」になったのだ。これはこれで正しい流れなのである。しかも，「いままでのイメージのパソコン」と同じアーキテクチャなのだから，断絶したわけではない。

では，第2期はどうなるか。ネットワークである。パフォーマ630はすでにビデオ・オン・デマンドのセットトップボックスとして使われることを念頭に置いている。いまでもFAXモデムとビデオキャプチャ機能があれば，シャープの液晶ビューカムなどなくても，ビデオ画像を通信で送るなんて簡単だ。こうしたコミュニケーションへの道を歩む。発信もできるメディアだ。

いまはまだネットワークも整備されていないし，スーパーハイウェイな時代も遠いが，目指すところはそこなのだ。そのために，パソコンを家庭に入れたい。いまのパソコン通信でネットワークのイメージを定着させたいし，CD-ROMソフトでパソコンをメディアとして使う用途を定着させたい。そのうえで，新たな展開を見せるのである。

おそらく，第2期の家庭向けマルチメディアパソコンは，ハードウェアの動画処理機能を持っているはずだ。これにより，ビデオCDだけでなく，さまざまな映像メディアをこなせるようになる。ビデオ入出力端子が前面について，もっと扱いやすくなるだろう。モニタが横長になっているかもしれない。通信機能ももっと充実するだろう。

たぶん，ビデオ・オン・デマンドなんてのが出てくるときは，第3期マルチメディアパソコンの時代になっているのではないかという気もする。

そんな時代，どんなOSがハシリ，どんなことができるようになっているか，ってなことを来月は考えたいと思う。

愛読者モニタ

モニタの応募方法

希望するモニタ記号をとじ込みのアンケートはがきの右下のスペースまたは官製はがきにひとつ記入してお申し込みください。応募の際に使用環境について明記する必要はありませんが，当選された方にはモニタとして使用ののちレポートを提出していただきます。締め切りは1994年12月18日の到着分までとし，当選者の発表は1995年2月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

A SX-WINDOW開発キット Workroom SX-68K

X68000用 5名

3.5/5″2HD版 39,800円(税別)

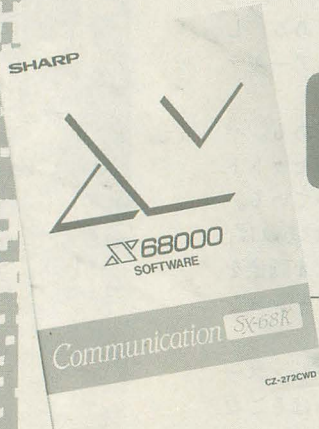
シャープ ☎03(3260)1161

B C COMPILER PRO-68K ver.2.1 NEW KIT

5名

X68000用

3.5/5″2HD版 44,800円(税別)

シャープ ☎03(3260)1161

D Communication SX-68K

2名

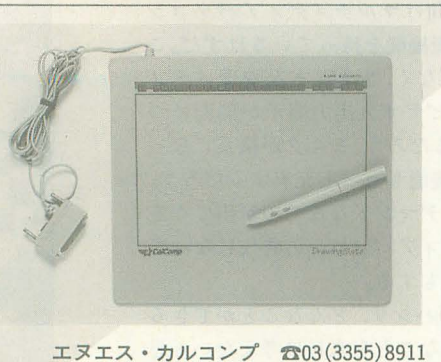
X68000用 3.5/5″2HD版 19,800円(税別)

シャープ ☎03(3260)1161

E Drawing Slate

1名

74,800円(税別)

エヌエス・カルコンプ ☎03(3355)8911

5 X68k Programming Series 各5名

A (#0) X680x0 Develop. & libc Ⅱ
2,900円(税別)

B (#1) X680x0 Develop. Manual Books
5,300円(税別)

C (#2) X680x0 libc Manual Books
6,300円(税別)

D (#3) X680x0 TEX
9,800円(税別)

ソフトバンク ☎03(5642)8100

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1994年12月18日の到着分までとします。当選者の発表は1995年2月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

C Easydraw SX-68K

2名

X68000用　3.5/5″2HD版　19,800円(税別)

シャープ　☎03(3260)1161

1 倉庫番リベンジ SX-68K ユーザー逆襲編

A 5″2HD版 2名

B 3.5″2HD版 1名

X68000用　各6,800円(税別)

シャープ　☎03(3260)1161

2 餓狼伝説 SPECIAL

3名

X68000用

5″2HD版　9,800円(税別)

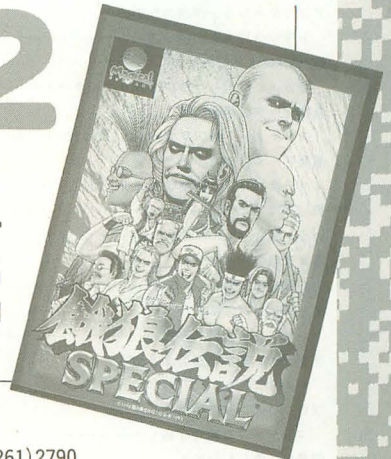
魔法株式会社　☎078(261)2790

3 FreeSoftwareSelection Vol.2

2名

X68000用

CD-ROM版　6,000円(税別)

計測技研　☎0286(22)9811

4 「ツクモDOS/Vパソコン本館」オープン記念 パソコンお掃除キット

10名

ツクモ電機

10月号プレゼント当選者

❶レッスルエンジェルス3　(茨城県)下平武彦　(岐阜県)田中敬久　(和歌山県)深見満彦　(京都府)平谷淳一　(香川県)津田直樹　❷最新MIDI DATA制作術　(群馬県)杉山和憲　(東京都)及川大之　(熊本県)千馬宏　❸「満開の電子ちゃん」　(埼玉県)進藤智文　(千葉県)小森達輝　(東京都)渡辺　誠　(長野県)原　真二　(愛知県)森　孝夫　❹「ツクモの日」記念テレホンカード　(群馬県)末岡丈明　(埼玉県)川井健司　奥村真明　(千葉県)中倉信吾　(東京都)国竹泰夫　(神奈川県)星野俊英　是枝浩行　(岐阜県)中野克己　(岡山県)西山完二　(広島県)田中栄治　❺新雑誌創刊記念テレホンカード　(茨城県)中内崇夫　(埼玉県)本田英雄　(神奈川県)金子満生　永井　孝　(静岡県)藤田康一　(三重県)辻本義成　(岐阜県)伊藤　治　(京都府)藤原直美　(徳島県)久米健司　(福岡県)島崎智博

(敬称略)

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

特集

特別企画

特別付録ディスク

THE SOFTOUCH

活用・レポート

連載・シリーズ

全機種共通システム

イベント/ギャラリー

製品紹介

INFORMATION

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

日本語カラーワープロ
WD-C700/WD-C700VS
シャープ

WD-C700

シャープはプリンタを内蔵した日本語カラーワープロ「WD-C700」「WD-C700VS」を発売した。

「WD-C700」はハイコントラストSTNカラー液晶を採用し，640×480ドットの表示を実現した。ワープロ部分では約130万語の辞書や12種類のアウトラインフォントを搭載している。ビジュアル文書の作成を助けるアート倶楽部などのアプリケーションは書院シリーズの特長を受け継いでいる。それぞれのアプリケーションはカラーに対応し，サンプルも数多く用意されている。

印刷部では解像度が400dpiの熱転写カラープリンタを搭載している。4色のインクリボンは色ごとに交換が可能で，コストの面で効率がよい。印刷アプリケーションも名刺印刷など10種類が用意されている。

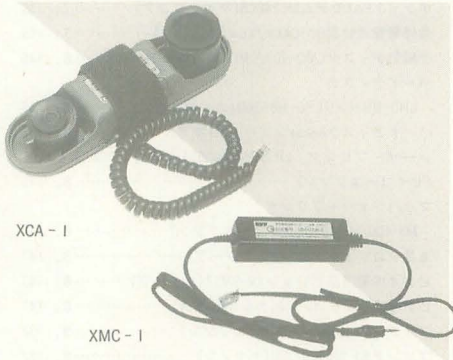
さらに，ザウルスや書院シリーズとのコードレス光通信，他社ワープロの文書データが利用できる他社文書呼び出し，約4.2万語の国語辞典機能などを搭載。

「WD-C700VS」は「WD-C700」の機能にビデオ入力アダプタとハンディカラースキャナを加えたもの。

価格は「WD-C700」が250,000円，「WD-C700VS」が308,000円（それぞれ税別）。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(5261)7271

音響カプラ/携帯電話接続コード
XCA-1/XMC-1
ビーピーエス

XCA-1　XMC-1

ビーピーエスは音響カプラ「XCA-1」と携帯電話用データ通信コード「XMC-1」を発売した。

「XCA-1」は，外出先で電話回線はあるがモジュラジャックがない場合のデータ通信を助ける。通信速度は300～14,400bpsに対応（回線の状況によってはそれ以上も可能）。ただし，モデム機能は内蔵していないので別途必要になる。使用電源はアルカリ9V電池を使用しており，連続で30時間の使用が可能になっている。重さは約210g。

「XMC-1」を使えばアナログ式携帯電話を使ってパソコン通信やFAXができる。電話側の接続にはイヤホン/マイク端子を使用。通信速度は300～7,200bpsに対応。

価格は「XCA-1」が14,800円で「XMC-1」が7,800円（それぞれ税別）。

〈問い合わせ先〉

㈱ビーピーエス　☎045(475)3726

ビデオ編集機
XV-AL300
ソニー

ソニーはビデオ編集機“ファミリースタジオ”「XV-AL300」を発売した。

同機は編集コントローラ，タイトラー，オーディオミキサー，CDプレーヤーを1台にまとめたビデオ編集機である。

XV-AL300

編集コントローラ部分では2つのモードが用意されていて，手軽に編集したり，プログラム編集が可能。CDプレーヤーの再生や停止などの制御はここで行う。

内蔵されたタイトラーはJIS第1/第2水準の漢字に対応し，文字の色は12色から，サイズは4種類から選択できる。また，文字に合わせた入学式などのサンプルイラストが約180種類登録されていたり，メニューで選択するだけで，タイトルのフェードイン/アウトやスクロールが行える。

ほかにも，オプションのマイクを利用してナレーションなども吹き込むことが可能。付属品はS映像ケーブル，映像/音声ステレオケーブル，ACアダプタ，著作権フリーのBGM用CDなど。

価格は78,000円（税別）。

〈問い合わせ先〉

ソニー㈱　☎03(5448)3311，06(251)5111

ビデオフォトプリンタ
SCREEN SHOT
日本ポラロイド

日本ポラロイドはビデオフォトプリンタ「SCREEN SHOT」を発売した。

同機は本体内の1.5インチCRT画面をそのまま露光するというCRT光学多重露光方式を採用した。このため，特殊な印画紙ではなく，自己現像方式の「ポラロイド・スペクトラフィルム」（1パック10枚，2,200円）を使用する。プリントサイズは67mm（縦）×91mm（横）で，1枚当たりのプリントアウトは出力/現像時間を合わせて約100

SCREEN SHOT

秒。操作は基本的に映像をメモリに取り込むキャプチャーボタンとプリントを開始するプリントボタンで終了。あとは必要に応じて明暗コントロールやメモリ画面呼び出しボタンなどを使用する。

価格は69,800円（税別）。

〈問い合わせ先〉

日本ポラロイド㈱　☎03(3438)8835

CD-ROMドライブ&ドライバ
CDS-E
満開製作所

CDS-E

満開製作所はメルコのCD-ROMドライブ「CDS-E」に計測技研社製の「CD-ROM Driver ver.2.00」を同梱して販売する。

「CDS-E」の動作モードはモード1（2倍速，転送速度360Kバイト/秒）とモード2（2.4倍速，410Kバイト/秒）の2種類がある。X680x0では10月末現在でモード2に対応していないが，同社でバージョンアップサービスを行う予定。平均アクセス速度は250msで，バッファ容量は256Kバイトを内蔵，インタフェイスはSCSI-IIに対応している。また，電動トレイ式なのでキャディは不要。

付属品はX680x0用デバイスドライバー式(CDPLAY,SXMACDIRなどのフリーソフトウェアも同梱)，SCSIケーブル(双方ハーフピッチ）など。

価格は29,800円（税別）。

〈問い合わせ先〉

㈱満開製作所　☎03(3554)9282

はがき印刷機
HV-50
カシオ計算機

HV-50

カシオ計算機ははがき印刷機“ポストランド”「HV-50」を発売した。

同機は解像度が360dpiの熱転写プリンタを内蔵している。フォントも毛筆楷書体と丸ゴシック体があり，連続30枚までの印刷が可能なオートシートフィーダを装備している。

宛名書きには住所録として約330件のデータを登録でき，入力も頭文字から都道府県や市区郡を呼び出せるようになっている。差出人も最大5人まで設定でき，住所録のデータも差出人別に登録することが可能。レイアウトについては住所の長さを判断して自動的に文字サイズや字間，配置などを変えてくれる。はがき裏面の内容は年賀状や暑中見舞いなどの5分野42種類の文例や7色カラーのイラスト15種類が用意されている。さらに，ビデオデッキと接続することで，自分で撮影した画像をイラストの代わりに印刷することもできる。

1枚当たりの印刷速度は宛名が約1分50秒で文面が約3分45秒（半面カラー印刷）。

価格は35,000円（税別）。

〈問い合わせ先〉

カシオ計算機㈱　☎03(3347)4830

英単語電子学習機
GK-E510
シャープ

シャープは英単語を学習するための電子学習機「GK-E510」を発売した。

同機は「英単語ターゲット1900」（宮川幸久著，旺文社刊）の内容をそのまま内蔵（英単語：約2,900語，熟語：約700語）した電子学習機である。学習方法には英単語，発音記号，意味を一覧してセットで覚える方法と，英単語だけを見て意味を覚える，意味だけを見て英単語を覚えるという3種類がある。また，覚えていない単語に印をつけて集中的に学習することも可能。テストモードは，単純に意味や英単語を当てたりする以外に穴埋め問題も用意されている。

GK-E510

価格は8,500円（税別）。

〈問い合わせ先〉

シャープ㈱　☎06(621)1221，03(5261)7271

INFORMATION

'94EPSON
カラーイメージング大賞
'94EPSONカラーイメージング大賞事務局

エプソン販売が協賛で「'94EPSONカラーイメージング大賞」を実施する。同コンテストの内容は以下のとおり。

応募期間：1994年10月17日
　　～1995年1月20日（消印有効）

応募テーマ：
　年賀状部門：平成7年年賀状
　一般プリンティング部門：自由テーマ
　動画部門：自由テーマ

応募条件：応募点数は応募者1名につき1点。未発表のオリジナル作品に限る。応募作品の著作権はエプソン販売に帰属する
　年賀状部門：使用する機材のメーカー，機種は問わない。ハガキサイズを使用
　一般プリンティング部門：同上。用紙はA4サイズを使用
　動画部門：フロッピーまたはCD-ROM。EPSON PCで動く5分以内の作品

賞金：グランプリ100万円ほか

応募方法：住所，氏名，年齢，職業，電話番号，部門名などを明記

詳しくは下記まで問い合わせを。

〈問い合わせ先〉

'94EPSONカラーイメージング大賞事務局

☎03(3563)5857(土・日除く10：00～17：30)

FILES Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。寒さの苦手な人は北風が身体にこたえていることでしょう。そんな日は家で鍋物を囲みながら仲間と騒ぐと心も身体も温まるかも。

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
電撃王　主婦の友社
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶NEWS
富士写真フイルムの100Mバイト以上のデータが入る3.5インチFD開発のニュースなど。――編集部，ASAHIパソコン，10・15号，8-11pp.

▶パリスの審判
架空ソフト発明レースの最終回。太陽系創造ゲーム「セカンド・ユニバース」などのアイデアが登場。――井上正基，ASAHIパソコン，10・15号，38-39pp.

▶EDUCATION
インターネットを学校教育に取り入れる試みを紹介する。今回は山梨大学教育学部付属小学校。――坂本伸之，ASAHIパソコン，10・15号，44-45pp.

▶ディスプレイとの交際術みつけた！
17インチディスプレイのなかから10機種を選んで紹介する。ディスプレイの規格や調整機能の解説つき。――編集部，ASAHIパソコン，10・15号，108-115pp.

▶機械用言博物館
最終回の今回は「シャットダウンする」をキーワードに，パソコンのソフトやハードの終了方法について触れる。――荻窪圭，ASAHIパソコン，10・15号，120-121pp.

▶アメリカ真夏のハッカー会議
ラスベガスで行われたハッカーのコンファレンス，デフ・コンに参加した筆者の体験談。――笠原利香，ASAHIパソコン，10・15号，124-125pp.

▶CD-ROM NEWSTYLE
単なるプラネタリウムでなく，星の1つひとつの軌道がシミュレートされている“箱宇宙”ソフト「RED SHIFT」などを紹介。――編集部，LOGIN，20号，38-39pp.

▶ハードウェアFLASH!
ロジテックのディスプレイ「LCM-15」やコパルのMOドライブ「PowerMO」など，ハードウェアの新製品情報。――編集部，LOGIN，20号，40-43pp.

▶特集1　テーマパーク上陸!!
あの「ポピュラス」のピーター・モリニューが制作したシミュレーション「テーマパーク」の魅力を解析する。――編集部，LOGIN，20号，123-135pp.

▶特集2　失敗しない!! CD-ROM選び
CD-ROMの魅力を紹介する。東京，大阪のソフトショップガイドつき。――編集部，LOGIN，20号，136-143pp.

▶ゲーム博物館，呉港を制圧!!
広島県呉市にオープンした「ゲーム博物館」を紹介する。懐かしのゲームがたくさん置いてある。――編集部，LOGIN，20号，170-171pp.

▶MCD-ROMの時代がやって来る!!
ミュージシャンの作成したCD-ROMを紹介し，その可能性を探る。――編集部，LOGIN，20号，178-181pp.

▶くねくね科学探検　第7回
NTT基礎研究所の光永正治氏などが開発している「ホログラフィック動画記録」について解説する。――鹿野司，LOGIN，20号，188-191pp.

▶特集　パソコン通信のすべて・初級編
パソコン通信を始めるためのガイド。通信サービスの内容，各ネットの特徴などを紹介する。――編集部，コンプティーク，11月号，25-36pp.

▶特報　次世代機ソフト最前線
PlayStation用「リッジレーサー」やSEGA SATURN用「バーチャファイター」などの進捗状況を報告。――編集部，コンプティーク，11月号，119-127pp.

▶NEWS COLLECTORS
任天堂32ビット機のベースシステムとなる「プライベート・アイ」の実態など，ゲーム業界のニュース。――編集部，電撃王，11月号，20-23pp.

▶特集1　次世代機はやっぱりソフトで選ぶ
PlayStation用ゲームの最新情報を紹介する。「リッジレーサー」「闘神伝」「ヴァンパイヤ」など。――編集部，電撃王，11月号，25-33pp.

▶A＆V環境をパワーアップしよう！
AV関連のパソコン周辺機器のお勧め製品をパソコン販売店のアドバイスを交えて，ピックアップする。ディスプレイや防磁型スピーカーなど。編集部，マイコンBASIC Magazine，10月号，35-43pp.

▶大容量化する記憶媒体たち　第1回
今回はヤノ電器の230Mバイトの3.5インチMOドライブ「MOBILSHUTTLE」を紹介する。――編集部，マイコンBASIC Magazine，11月号，48-49pp.

▶ヤマハが新音源規格を提唱！
GM音源をさらに拡張したXG音源。その第1号となるヤマハの「MU80」を取り上げてその内容に迫る。――編集部，マイコンBASIC Magazine，11月号，54-55pp.

▶新ハード特捜部
PlayStationとSEGA SATURNなどの最新ソフト情報を公開。――山下章，マイコンBASIC Magazine，11月号，147-152pp.

▶Arcade Game Graffiti
1981年のアーケードゲームを振り返る。「ボスコニアン」「ギャラガ」「ムーンシャトル」などを紹介。――編集部，マイコンBASIC Magazine，11月号，164-167pp.

▶NEWS
「データショウ'94」のレポート，第7回アマチュアCGAコンテスト作品募集のお知らせなど。――編集部，ASAHIパソコン，11・1号，8-11pp.

▶見えた！　情報スーパーハイウエーの全貌
ホワイトハウスの情報テクノロジー担当特別補佐官マイケル・ネルソン氏など，情報スーパーハイウエーに関わる人々へのインタビュー。――室謙二・西村由美，ASAHIパソコン，11・1号，14-19pp.

▶CD-ROM99マラソン
数あるCD-ROMソフトの中から，ASAHIパソコンのお勧め99枚を紹介する。また，CD-ROMソフトの制作現場をレポートする。――編集部，ASAHIパソコン，11・1号，20-33pp.

▶パソコン通信が変わった！
パソコン通信の新たなトレンドを追跡する。GUIを採用した通信ソフト，インターネットとの接続など。――山田祥平ほか，ASAHIパソコン，11・1号，110-119pp.

▶98ユーザーのためのマッキントッシュ教室　1
98ユーザーがマックを使いこなせるようになるための教室。1回目は「電源スイッチはどこにある」。――荻窪圭，ASAHIパソコン，11・1号，132-135pp.

▶アメリカ・ハッカー会議報告
会議での催しのひとつ「第2回ウイルス・コンテスト」から「人の役に立つウイルス」について考える。――笠原利香，ASAHIパソコン，11・1号，136-137pp.

▶新世代α徹底研究
今年後半から来年にかけて次々にバージョンアップするOSを検証する。「Windows95」「OS/2」「System7.5」が登場。――編集部ほか，I/O，11月号，33-62pp.

▶大容量記憶装置
大容量のメディアの現状をまとめる。MO，HDのQ＆Aと最新製品の紹介。――田嶋孝行，I/O，11月号，78-91pp.

▶個人輸入に挑戦！
実践的な個人輸入の方法やノウハウを紹介。――山本雅章，I/O，11月号，105-108pp.

▶Multimedia Watching 11
今回は「日本で成熟しつつあるメディアとは」と題して，ポケベルやワイドテレビなどを取り上げる。――奥野雅之，I/O，11月号，130-133pp.

▶C言語教室
構造体について学習する。「任意の年月日の曜日が求められるプログラム」がサンプル。――関口智宏，I/O，11月号，147-152pp.

▶特集I　後期カラーノート主義
カラーノートパソコンのトレンドを解説し，各マシンをレビューする。ベンチマークテスト結果とスペックの総覧つき。――編集部，ASCII，11月号，261-288pp.

▶特集II　プライベートCD-ROMを作ろう！
自作CD-ROMソフトを作る。実際にいくつかのソフトを使って試してみる。――編集部，ASCII，11月号，297-312pp.

▶PRODUCTS SHOWCASE
EPSONのスキャナ「GT-6500WIN2」，キヤノンのカラープリンタ「BJC-400J」などの新製品を批評する。――編集部，ASCII，11月号，328-343pp.

▶新科学対話　第11回
東京大学法学部教授の石黒一憲氏と，NTT基礎研究所の竹内郁夫氏が高速ネットワーク通信の現状と未来について対談。――編集部，ASCII，11月号，344-350pp.

▶稀代もののけ考

ハイテク関連のちょっと変わったグッズを紹介。——バカパパ, ASCII, 11月号, 440-441pp.
▶なんでも相談室
CDのデータを使ってTシャツやトレーナーを作るにはどうしたらいいか。熔融型プリンタやショップのサービスを紹介。——編集部, ASCII, 11月号, 452-455pp.
▶特集2　カラーイメージスキャナ活用研究
いくつかのスキャナを実際に活用してみる。X68000を利用したシャープ「JX-330」のパフォーマンステストほか。——編集部, My Computer Magazine, 11月号, 25-37pp.
▶パソコン研究室
パソコンで音を捉えるときのメカニズム, PCMについて解説する。——Space Club, My Computer Magazine, 11月号, 114-116pp.
▶ビジネスマンのための情報管理術
シャープの携帯情報端末, ザウルスの活用講座。ザウルスに対応したパソコンソフトを概観してみる。——塚田洋一, My Computer Magazine, 11月号, 126-127pp.
▶HardwareForumX
大容量ハードディスクと高速モデムのチェックポイントと製品の紹介。——編集部, LOGIN, 21号, 166-169pp.
▶次世代ゲーム機最終情報報告
各ハードのスペック紹介と, 各社の意気込みを聞く。——編集部, LOGIN, 21号, 192-197pp.
▶ロンドンゲーム紀行
ヨーロッパのゲームの祭典「'94ECTS」を通じてヨーロッパゲーム業界の動向を探る。最新ソフトが目白押し。——編集部, LOGIN, 21号, 198-203pp.
▶人間VSモザイク100年戦争
モザイクをかけた画像データを100%もとの画像に復元することに挑戦する。——編集部, LOGIN, 21号, 208-211pp.
▶くねくね科学探検　第8回
今回は世にあまり知られていない科学者テレミン氏とテスラ氏を紹介する。——鹿野司, LOGIN, 20号, 188-191pp.

X1/turbo/Z

X1シリーズ
▶EGG CARRIER 2
卵を船に運び込む, 思考型パズルゲーム。——大前弘, マイコンBASIC Magazine, 11月号, 114-116pp.

X68000

▶Release Data
各機種別ゲームの発売日情報。EAVの「魔法大作戦」など。——編集部, LOGIN, 20号, 8-9pp.
▶SUPER SOFT INDEX
発売予定のソフトを掲載。X68000用はネクサスインターラクトの「VIEW POINT」など。——編集部, コンプティーク, 11月号, 107-110pp.
▶電撃新作予定表
X68000用にはEAVの「上海　万里の長城」, TAKERUの「レッスルエンジェルスSPECIAL」などが登場。——編集部, 電撃王, 11月号, 165p.
▶The Shooting(射撃)
画面に表示される的にマウスカーソルを合わせてクリックするアクションゲーム。——服部誠, マイコンBASIC Magazine, 11月号, 117-118pp.
▶人工知能ツクールVer1.00
人工知能を自分で作り, 実行するためのツール。——びんぼうくん, マイコンBASIC Magazine, 11月号, 119-121pp.
▶ネビュラスレイ～Exeo～
NAGDRV用ミュージックプログラム。——小島真志, マイコンBASIC Magazine, 11月号, 128-129pp.
▶SUPER SOFT HOT INFORMATION
X68000用はビデオゲームアンソロジーVol.11「パックランド」が登場。——編集部, マイコンBASIC Magazine, 11月号, とじこみ付録12p.
▶AV STRASSE
IMAGICAテクノシステムが発売するX68000用レンダラー「XL/Image」を紹介。——編集部, ASCII, 11月号, 389-392pp.
▶ON-LINE SOFTWARE INDEX
大手主要ネットにアップロードされたソフトを紹介する。X68000用は多機能高速ファイル検索ソフト「qw.x version0.41」など。——編集部, ASCII, 11月号, 477-483pp.
▶なんでもQ＆A
「SX-WINDOW ver.3.1」のシャーペンに付属している外部コマンドmapの使い方などの質問に答える。——編集部, My Computer Magazine, 11月号, 153-155pp.
▶スターラスター
マイコンソフトの「スターラスター」のレビュー。——あゆさわかつみ, My Computer Magazine, 11月号, 172p.
▶SX-WINDOWプログラミング
シャーペン用の外部コマンドとして“簡易C文法チェッカ”を作成する。今月はプロトタイプを作成。——吉野智興, C Magazine, 11月号, 118-127pp.

ポケコン

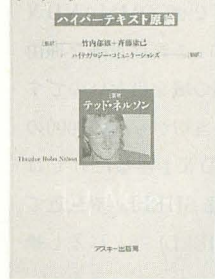
PC-E500
▶イケイケゲーム
壁を避けて, JUMPしながら進むアクションゲーム。——稲葉大祐, マイコンBASIC Magazine, 11月号, 122p.

新刊書案内

電子図書館
長尾　真著
岩波書店刊
☎03(5210)4000
四六判　125ページ
1,000円（税込）

最近岩波新書でベストセラーを出した電脳関係科学者といえば,『マルチメディア』の西垣通と『人工知能』の長尾真だ。特に長尾真はかの『岩波情報科学辞典』なる大著も手がけたほどである。その長尾教授が岩波科学ライブラリーで「電子図書館」を著した。この流れからわかるとおり, 電子図書館とは, マルチメディアとネットワークによって作られるオンライン図書館のこと。そこには電子化された図書が納められており, ハイパーテキストの要領で自在に目的の本を検索することができる。ペーパーメディアで流通していた本はOCRなどを使ってデジタル化され, 目次や抄訳がデータベース化され, 検索に用いられる。論文などはそのまま登録され, 書物の枠を超えたアクセスが可能になっている。その図書館は, ユーザーが自分で操るほか, 図書館員がリクエストに応じて利用者の求める情報を検索するサービスも行う。現在インターネット上で行われている世界を図書館というメタファーでもって, より現実的に構想し直しつつ, それを誰でもわかるように解説した本なのだ。図書館化は「情報に構造を導入することによって高度な検索を行い, 柔軟性のある情報提供を行う」ものであり, ユーザーが積極的に「システムにかかわっていく必要がある」インターネットとの違いがここにあるわけだ。
電子図書館は著者が研究・開発を進めているシステムであると同時に, 情報スーパーハイウェイによるインフラを前提とした, 極めてわかりやすい利用法である。図書館をまるまる電子化するとどのくらいの容量が必要なのか, 検索用の「構造」をどのように作っていくか, それに伴い, 現在の図書館というシステムはどうなっていくか。著作権の問題もある。そう考えると電子図書館というのは非常にわかりやすくてよい。問題は, 多くの人が図書館に対してあまりにクールであることにあるかもしれない。　(K)

知の探偵術
武田　徹著
PARCO出版刊
☎03(3477)5755
A5判　249ページ
1,600円（税込）

皆さんは必要な情報を手に入れるために, どういった方法をとっているだろうか?
ちまたには情報収集のためのマニュアル本が溢れているが, 本書は単なる情報収集のためのマニュアル本ではない。著者が実際に情報社会のなかでの情報収集を行った, 主観によって統一された個人的な調査レポートである。調査している項目は, 図書館, インタビュー, アンケート, 考現学, うわさ, パソコン通信など。その調査をとおして, その情報収集方法の問題点を明確にしていき, 解決方法を考えていく。
本書を生かせるかどうかは, あなた次第である。

リテラシーマシン
テッド・ネルソン著
ハイテクノロジー・コミュニケーションズ翻訳
竹内郁雄＋斉藤康己監訳
アスキー刊
☎03(5351)8194
四六判　406ページ
2,800円（税込）

本書はあるプロジェクトの報告書である。その名前はザナドゥ・プロジェクト。そこで展開されることは, 電子出版, ハイパーテキスト, 思考のための玩具, 未来における知性の革命, さらに知識と教育の自由といっている。
著者はワードプロセッシングの発明者のひとりともいわれるテッド・ネルソン。本書も基本的には彼が1974年に自費出版したものである。ただ, 20年も前に書かれたにも関わらず, その内容は現代の各方面にいろいろな影響を与えている。
すべてを鵜呑みにする必要はないが, なんともいえない怪しさに興味をそそられる1冊である。

QUESTION and ANSWER

こんな質問をここにしてよいかわかりませんが，Human68kのformatコマンドの仕様で，どうしても納得がいかないことがあるので，こうしてお手紙しました。

実は私はMS-DOSも使っています。MS-DOSのFORMATコマンドって，/Sオプションを使ってフロッピーをフォーマットすると，IO.SYSとMSDOS.SYSのほかにCOMMAND.COMもコピーしてくれますよね？　ところがHuman68kでは/SオプションではHUMAN.SYSはコピーするものの，COMMAND.Xはコピーしてくれません。

これってどうしてなんでしょうか？

北海道　山下　滋

ええと，これだけ読んだだけではわからないことがあると思うので少し補足しておきましょう。

まず，FORMATコマンドの/Sオプションというのはフォーマット後にシステムファイルもコピーするというものです。

次にIO.SYS，MSDOS.SYSですが，これはどちらもMS-DOSのシステムファイルです。Human68kではHUMAN.SYSに相当するものですね。COMMAND.COMとCOMMAND.Xの関係はいわなくてもわかりますよね。

さて，Human68kでなぜCOMMAND.Xをコピーしないのか？　これは正確な理由はわかりません。推測の域を出ないのですが，Human68kが出た当初から，X68000のSHELLはCOMMAND.X 1種類だけではなく，VS.X (VISUAL SHELL) や最近ではSXWIN.X (SX-SHELL) などに差し換えるという想定があったからでしょう。

だからといって，いまさらフロッピーディスクにVS.Xをコピーしたがる人はあまりいないでしょうし，SXWIN.Xをフロッピーディスクで動かそうなんて人もそんなにいないでしょうから，COMMAND.Xを/Sでコピーしてほしいと思うのは無理のない話かもしれません。

そのためにはどうすればいいのでしょうか？　実は，これは簡単に実現できます。というのも，/SオプションはHuman68kのHUMAN.SYSをコピーするものではなく，ルートディレクトリにあるシステム属性のファイルすべてを同時にコピーするというオプションだったのですね(Human68k ver.3で確認)。

ということは，ATTRIB.Xコマンドでもって，ルートディレクトリのCOMMAND.Xをシステム属性にしておけばよいというわけです。

具体的には，

A>ATTRIB +S COMMAND.X

とすればCOMMAND.Xのシステム化が実現できます。

こうしておけば，/SオプションでCOMMAND.Xもコピーしてくれるので，便利な人には便利でしょう。同様にKEY.SYSやBEEP.SYSだってシステム属性にしておけば自動的にコピーしてくれるという優れものです（まあそんなものまでシステム属性にする人はいないだろうけど）。ただし，システム属性というのはOSにとって重要な意味を持つものですから，無闇に指定するのはあまりおすすめできませんが。

ところで，Human68k ver.3.0以降の，FORMATコマンドには，マニュアルに記載されてないようなオプション指定項目がいくつかあります。

/4　2HD1.44Mバイトのフォーマット
/5　2HCのフォーマット
/8　2DD640Kバイトのフォーマット
/9　2DD720Kバイトのフォーマット

これらの指定はSX-SHELL上では簡単にできましたが，実はCOMMAND.X上でもできたんですね。なお，これらはすべてFDDEVICE.Xが常駐されているときにしか利用できません。FDDEVICE.Xの表記メッセージのとおり，/5以外はツクモの3モード対応増設ドライブしか利用価値はないでしょうけどね。　（瀧　康史）

コンソールがサポートされたので最近はSX-WINDOW上でシャーペンを使ってプログラミング作業をしています。エディタ環境は快適なのですが，シャーペンでプログラムを記述してからいちいちウィンドウを切り換えてコンパイルするのはちょっと面倒です。なにかいい方法はないでしょうか。一時はサンプルメイクを使っていましたが，それもイマイチ使い勝手がよくありません。シャーペン上からコマンド1発でコンパイル……とかできるとよいのですが。

静岡県　大山　良典

プログラム開発環境も徐々にSX-WINDOWに移行している人が増えています。しかしコンパイラなどはSX-WINDOW用に作られているわけではありません。そこでシャーペンでサポートされたコンソールを開いて作業することになるのですが，残念ながら現時点ではコンソールとのコミュニケーション手段が用意されていないようです（少なくとも公開では）。

ちょっと見たところでは，コンソールウィンドウ内はまわりのSXの環境とは隔絶していますので，タスク間通信などでデータをやりとりしたりすることはできそうにありません。

しかし大山さんと同様のことは私も以前から考えていました。要はSX側からのなんらかの合図をキャッチしてコンソール上のプログラムが動作できればよいわけです。そこで，けんと君に環境変数を書き換えるシャーペン外部ファイル作成を依頼してみたりと，いろいろ検討してみたのですが，外からコンソールの環境に手を加えることはできそうにありませんでした。

いろいろ試してみて最近落ち着いたのはテンポラリファイルを使う方法です。

まず，RUNMAKE.BATとしてリスト1のようなバッチファイルを用意します。次にシャーペンのキー定義を書き換えます。デフォルトではOPT.1+Mが空いていますので，これをmakeコマンドにしてしまいましょう。キー定義ファイルの該当部分を，

#87,$0d,'I:¥TMP.$$$',$d

のように設定しておきます。

ここで指定するファイルのパスはどこでもかまいませんが，できればRAMディスクなどのアクセスの高速な場所にしておいてください。もちろん，リスト1との整合を取ることは忘れないように。

これでシャーペンの環境を再設定すれば

下準備は完了です。

コンソールを立ち上げて，カレントディレクトリをソースプログラム（というか，makefile）がある場所に移し，

A>RUNMAKE

を実行してください。

あとはシャーペンでソースプログラムをエディットし，コンパイルしたくなったらOPT.1＋Mを押すことでコンパイル処理が実行されます（直前にテキストをセーブしておくこと）。

ある程度バッチファイルの知識がある方なら，見ればだいたいなにをやっているかはわかるとは思うのですが，シャーペン側は合図として特定のファイル名のファイルを書き出します。一方，コンソールのバッチファイルはそれを見張っており，そのファイルがあったら特定の処理を行うようになっています。処理が終わったら合図に使ったファイルは消しておきます。

シャーペン側の指定でファイル出力時にファイル名を与えている部分などは参考になるかもしれません。

もちろん，コンソールで実行されるのは単なるバッチファイルですからMAKEの代わりに希望の処理を書いておけばTeXのコンパイルとかその他さまざまな処理が実現できます。コンソール.CENVをちょっといじってやればコンソールのエラー発生行から直接タグジャンプさせることなどもおそらく可能でしょう。応用範囲はかなり広いので，各自で研究してみてください。

……とか書いていたら，いきなりけんと君が専用の外部コマンドを作ってきました。そのうち掲載できると思いますがそれまではこのバッチファイルで対応してください。

リスト1

```
echo off
rem  runmake.bat***
:a
if exist i:¥tmp.$$$ goto b
goto a
:b
make
del i:¥tmp.$$$
goto a
```

えーと，Z-MUSICでプログラムを打っています。10月号のグラディウス2のプログラムの中で「'」というのが出てくるのですが，これはどうやって出すのでしょうか？　ちなみにZ-MUSIC ver.2.0のマニュアルにはこのコマンドの解説もないようですが……。確か，強制キーOFFだったと思うのですが。

それともうひとつ。たまに，なにかのキーを押しながら電源を入れると（起動すると），なにか特定のイベントが発生するという記事が載っていますが，結構いっぱいあるようなので一度，一覧表でも作ってもらえるといいのですが。栃木県　古橋　康宏

マニュアルの本文には解説が抜けていますが，裏表紙の記号一覧では一応，強制キーOFFとして載っているはずです。

これは，もともと隠しコマンドのようなもので，ver.1.0を作っているときに確か進藤君あたりがどうしても必要とかいって無理やりつけさせていたような……。

機能はもちろんそのトラックがアサインされているチャンネルの強制キーOFFです。チャンネルに対して動作しますからトラックとチャンネルの関係を理解していないと誤動作のもとになるかもしれません。注意して使ってください。

なお，マニュアルにないのは単なる記載漏れのようです。同様の経緯から追加されたコマンドは適切なMMLコマンドがなくてかなり変な記号に割り振られていたりします。ver.1.0の頃のFM音源ノイズモードとかは最初は半角カタカナの「ノ」に割り当てられていましたし……。ほかにもZ-MUSIC ver.2.0初版で記載漏れがあって再版分から追加されたコマンドに，

?a,d

トラックワークへのデータ書き込み

というのがありました。そのトラックのワークaにデータdを書き込みます。値によっては暴走しますので十分理解したうえで使用してください。

次の質問にいきましょう。

X68000は起動時にキーボードの特定のキーを押すことによりさまざまなモードを設定することができます。もっとも有名なのがOPT.1キーを押しながら起動するとFD0からの強制起動になるというものでしょう。ハードディスクユーザーがゲームなどをするときには必須といえる機能でした。

古橋さんが挙げているCLRキーによるSRAM初期化はX68000SUPER以降に新設されたもので，それ以前の機種では無効です。

また，HELPキーを押しながら起動すると起動メニューが現れ，ハードディスクの起動パーティションを指定できます。これも場合によっては重宝するかもしれません。

通常のX68000ではXF1からXF5までを押しながら起動するとキーボードLEDの輝度を変えることができます。ただし，これはX68030では変更されており，XF1で10MHz相当のウエイト，XF2で16MHz相当のウエイトをかけて起動するという意味になります。XF3からXF5まではそれぞれ10MHz/16MHz/25MHzのウエイトでROM内のHuman68kを起動するX68000互換モードになっています。

役に立つ機能というのはこれくらいのものです。そのほか，ファンクションキーのF1～F3を押してキーボードをリセットすると（ケーブルの抜き差しとか），クリスマスツリー（もぐら叩きともいう）といった小技がいくつかありますが，あくまでも遊びのものですので，あまり役には立ちません。　（中野　修一）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

眠りから覚めても布団から出るのが辛い季節になりました。でも，クリスマスに忘年会，この1年を締めくくるイベントが盛りだくさんで寝てる閑なんかありません。さあ，寒さなんてどこかへ吹っ飛ばしてしまいましょう。

◆いままで，SX-WINDOWの壁紙にコレだ！ というものがなく，しかたなく4色データにコンバートして使っていました。6万色データもいいのですが，画面が4分割されるのがどーしても気にいらないし……。てなわけで，Xがぐるぐるまわるのには大変感動しました。無機質な感じがグレーを基調としたSX-WINDOWにピッタリ。あぁ，かっちょええのおー。

鹿又　健(25)東京都

今度は自分でお気に入りの壁紙動画を作ってみませんか。

◆MJ-700V2Cのカラーコピーはよかった。“ミナモアンシンシテMJ-700V2Cカウヨロシ”あと“JX-330Xアルトモットヨロシネ”

島田　尚司(27)新潟県

JX-330Xなんて贅沢はいわなくても，なにかイメージスキャナがあると，なかなかいいですよね。

◆「TeX入門講座〜てふてふらてふ〜」がいい！ 私は通信をやっていないので，TeXというのは名前くらいしか知りませんでしたが，記事を読んでみるとなかなか面白そうで，私も使ってみたくなりました。いままでは，ワープロとTeXの違いがわかりませんでしたが，全然別ものなんですね。

野田　敏之(23)神奈川県

TeXにせよワープロにせよ，それぞれ得意なところと不得意なところがあります。書くものに合わせてどちらかを選べばいいと思います。

◆次号予告のところに何度か「H.A.R.P」のことが書いてあったのですが，新製品紹介のところに一度も載っていません。早く載せてください(そのときは，H.A.R.Pのみとメモリボード使用時の結果をお願いします)。

岩本　純徳(21)千葉県

やっと今月号で紹介できました。残念ながら，メモリボードは到着せず，来月以降となります。それまでお楽しみに。

◆2カ月間放ったらかしだったJX-330Xをセッティングしました。X68000ACE君とX68000CompactXVI君をつなげるのに手間取り，試行錯誤を繰り返した結果だったので嬉しいです。シロートの私がX68000ACEをX68000CompactXVIの外付け5インチドライブになんて思ったのが間違いで，いまだにキーボードはいったん抜いてやらんとだめです。でも，自分ひとりでできたことに意義はあるでしょう，たとえ2カ月かかったとしても。それとJX-330Xは爆速です。

大谷　育子(26)大阪府

編集部にあるX68030CompactもX68000初代を外付け5インチドライブとして使っているのですが，やはりキーボードはいったん抜いてやらないとだめです。どうにかならないものですかね。

◆X-BASICで，ひいきにしているアーティストの歌詞カード検索プログラムを完成させました。小粒でも実用的なんですよ。とはいっても，そこはBASIC，余計な処理が致命的な遅さになって返ってきます。ない知恵をしぼってのささやかな高速化，これが熱いのです。C言語もいいけれど，X-BASICは絶対におもしろいと思います。

永井　邦彦(25)愛知県

自分でなにかを作る喜びは忘れられないものです。次はBASICコンパイラに挑戦してみるのもいいかもしれませんね。

◆物心ついたときにはコンピュータがあった，という世代ですから，コンピュータを既知の事実として受けとめている人も多いでしょう。しかし，キーボードもモニタも初めから存在していたわけではなく，現在の形になるには歯車式計算機からの長い道のりがあったわけです。と，しみじみ思わせてくれるのが幕張のIBMショウルームです。およそコンピュータというよりは工作機械といった感じのパンチカード式計算機の前であなたはきっと立ちすくむでしょう。

松永　貴輝(23)大阪府

たまには時代の流れを感じてみるのもいいかもしれませんね。

◆TeXの読み方について書いていた人がいましたが，正しい読み方のわからないコンピュータ用語は結構多いですね。「SASI」「SCSI」は有名ですけど，「GNU」とか「NEmacs」とかは何て読むのでしょうか？ ちなみに私は「ジーエヌユー」「ネマックス」と読んでいますが……。

小沢　基一(21)東京都

前者は「グニュー」で，後者が「ネマックス」とか「エヌイーマックス」と読むようです。

◆メモリが増えたので，シャーペンを操作していると，ついついディスク1枚ではとても収まらない文書を作成してしまいます。これが楽しいので，しばらくこの状態が続きそうです。そこにxclickがきたもんだから，アイヤーとばかりにいろいろな音を放り込んでいます。あぁ，なんて低級な遊びでしょう。プログラムの作成が……。

大久保　敦(21)大阪府

フロッピーディスク1枚に収まらない文章って……。大変な力作ですね。試しにその内容を送ってみませんか？ あ，入らないのか。

◆家で仕事に使っているX68000は，今度プログラムの大幅書き換えが行われることになった。それにともなって，1Gバイトハードディスクと X68000に接続される液晶ビューカムが家にきた。ちなみに家は美容院で，従業員の給料やお客様のDMはすべてX68000が処理しています。

市川　博基(19)愛知県

ということはお客さんの顔入りのデータベ

▲岡村　直也　兵庫県
いかにも初心者にありがちな失敗ですよね(本当か？)。でも，彼女の気持ちを考えると笑ってしまうのはかわいそう。健気でいいですね。

▲吉田　淳一　宮城県
おしゃれなPRO子もなかなかいいと思うのですが，衣装(アクセサリ？)を揃えたのは主様なのに……ちょっとひどくないですか？

ースでも作成されるのですか。ひょっとしたら，お客さんの顔にこの髪型だと……みたいにモンタージュでも作られるのでしょうか？

◆すいません。家のディスプレイはなまけてました。一応気合を入れなおしたけど，それでも816×544……。　小林　敦(19)埼玉県

あまり無理をさせると今度は寝込んでしまいますから，ほどほどに。

◆「餓狼SP」のレビューに載っていたコネクタの材料は，仙台地方ならダイエー仙台店の6階に全部そろっています。私もそこで買って作りました。マウスもバラして，PC-9801用(980円)をつないでいます。私のような素人でもできました。ところで，個人ユーザー向けのパソコンショップを特集してほしいです。ソフトの割引き，個人でも行きやすい穴場など，県ごとにぜひお願いします。私はOAシステムプラザ仙台店に行っています。「ストII'」，「スーパーストII」などの注文も全部そこでしています。　西條　勝(21)宮城県

「ダイエー恐るべし！」という感じですね。世界の王を監督に迎えたし，まさにこれからって……違う話ですね。ごめんなさい。

◆「満開の電子ちゃん」が夏コミで販売されていたなんて，気がつきませんでしたよ。岡村祭さんもどっかに書いておいてくれれば，買いに行ったのにー。サークル名はなんだったのですか？(カタログを見たけど見つからなかった)。冬コミで販売するのなら絶対に買いに行くので，ぜひ教えてほしいものです。　磯崎　玄(19)千葉県

はっきり決まったかどうかは不確実な情報ですが，電子(でんこ)出版から冬コミにまた別物が出るという噂です。

◆え！「XL/Image」が出るの？「きぇー」と電車の中で叫びそうになってしまいました。「MIRAGE Attribute stuff」が出ないと決まった時点で，「もーだめかな」と思っていたら，「XL/Image」ですか。そりゃ凄い，凄すぎる！　できればWindows版でいいからL/Shapeがあれば……などとゼイタクなことを考え，さらに「セットで15万くらいなら，考えてもいいなー」なんてね。X68030がちょっとほしいかなー。でも，Windows版出してくれればそれでいいし。でもほしい。よし，XL/Image基金を作ろう！　安井　百合江(20)愛知県

今月のお試し版はいかがでしたでしょうか。お眼鏡にはかないましたか？

◆約5カ月ぶりに日本へ帰ってきて，X68000をいじっていたら，ハードディスクが怪しい。さらに本体はもっと怪しい。突然止まるし，ゲームが動かない。修理をしたいが改造(青LED)してるから，ちょっときつい。「スーパーストII」をやりたいけど10MHzで2Mバイトしかないし……。　古橋　康宏(19)栃木県

ここはひとつX68030を買っておくのがよろしいかと……。

◆今朝，新聞を見て驚いた。「パソコンのモニタにテレビの映像が映るなんて，5年前には誰も予想しなかったスゴイ機能です」。アップルまでこんなことをいい出すとは。黙っていていいのか，シャープ！　竹内　周作(22)大阪府

きっと答えを出してくれる……といいな。

◆お腹の調子いまひとつだし，もともと胃腸は弱いほうなので，整腸剤なるものを飲みだしたら，結構よくなってきました。次はやっぱり養○酒かな。　松本　勝正(20)富山県

体は大切にしたいものです。でも，調子が悪くならないとなかなか気をまわさないですよね，困ったものですが。

◆某公務員の人物試験(個別面接)にて。「好きな学科はなんですか？」「はい，数学です。趣味でパソコンを使っていて，数式などは技術の応用にも役に立ちますので」「では，どんなパソコンをもっているのですか？」「シャープのX68000という，世界で最もクリエイティブなマシンです」。……私は嘘をいってませんよね，ね，ね？　小島　康(19)宮城県

自分のいったことに自信をもちましょう。誰がなんといってもとりあえず自分だけは信じられますから(本当かな？)。

◆私は父に2つめのアパートを建てたときの話を聞いた(なんと鉄之助は大家の息子だった……。借金のカタマリを背負っているともいえるが)。そのときにアパートの名前をつけねばならないという話になったが，父が出した案は「Castle Plain」……訳して……平野城。私の父のネーミングセンスの悪さは18年前から変わっていない。　平野　鉄之助(18)長野県

日本の会社のネーミングも似たようなものだと思います。プリンタで有名なあれや，タイヤで有名なそれとか……。

◆大学3回生の秋にしてついに電話を引きました。当初は「恋人と愛を語らう」つもりが3カ月以上前にその相手とは別れ，気がついたら電話を引いた翌日に話がまとまり，近日中にモデムがやってきてしまうようです。現在，バイトをしていないのですが，いったいモデムが来てしまったら生きてゆけるのでしょうか……。ひとついえるのは，愛は当分語らえないことですな。知人Tのばかーっ！(でもありがと)。　嶋村　謙(21)大阪府

最近はパソクラなんていうのもあるみたいですし……。がんばってそちらで新しい恋を探してみてはいかがでしょうか？　男女比率を考えるとちょっとつらいかな。

◆9月20日に20歳のイチローが200本安打達成。シーズン中にあちこちの新聞・テレビで「すごいぞイチロー」「がんばれイチロー」の大合唱。なぜだか非常に勇気づけられた私は二浪はすまいと誓う予備校生でした。　高畑　健治(18)兵庫県

来年は「ジロー」が活躍するといいですね(冗談)。

◆9月22日，給料日，その日は昼前からすごい雨だった。2時頃，増水で水が膝まで達し，私の車は床上浸水。でもなんとかエンジンがかかり脱出成功！　隣の先輩の車は……成仏してください。というわけですごい雨だった。下水も上がってくるし，タイヤとかバンパーとか流れてくるし(近所に車屋がある)。今度車を買うときはRVにしようとつくづく思いました。　境　直忠(22)宮城県

大雨のニュースをテレビでやっているとき，車が走っているシーンが映るのですが，走るというよりは，まさに泳ぐという感じです。車の中に水は入ってこないのでしょうか？

◆友人がいうには，ファミコンって「ファミリーコンプレックス」の略だそーです。じゃあ，パソコンとか，マザコンとか合コンはいったいなんの略だというのだろーか？　やっぱ，マザコンは「マザーコンピュータ」になって……。　森谷　好雄(17)北海道

そうするとパソコンは「パソピアコンプレックス」のことで……。でも「パソピア」ってどのくらいの人が覚えているか，ちょっと心配。

◆ある日曜日の朝6:30ごろ，天気は雨。ドライブの帰り道で前方に車らしき物体を確認。近づいてみると，それはぐっちゃんぐっちゃんになった事故車がひっくり返って，あわれカメのようになっていたのでした。ヒェー。自分の持っていたカサを，そこにいた人に渡してすぐその場

◀加藤　信夫　宮城県
発売からずいぶんたつが，いまだに根強い人気を誇る「R.C.」。限られた条件のなかで，最強のロボットを創造していく。素晴らしいですよね。

◀鈴木　道明　埼玉県
鈴木さんの描く女の子はかわいいのですが，それ以外のお茶目なキャラクターもいい味出してますよね。あと，これも移植希望なのかな？

を離れましたが，いやーびっくりした。で，帰ってみると渡したカサは妹の。私はブーブーいわれました。人助けしたのに……トホホ。

片倉　純也(20)宮城県

その場に残された人は事故の連絡はできていたのでしょうか？　そうでなければ，雨の中ひたすら立ち続けて……。

◆ゲーセンで「雷電DX」に凝っています。とても，おもしろいです。シューティングゲームの苦手な人は嫌うかもしれません。まだ初級コースでやっています。ゲーセン友達は初級コースで924万点。とても抜けません。どうしたらいいのでしょうか？　まずはレーダー破壊率100%を目指します，では。日比生　雅治(21)大阪府

「雷電DX」は実は練習ステージが超燃え！隠しキャラを壊し続けると難易度は上級より難しくなるんだな。ただいま練習ステージで気合を溜めるのに挑戦中。どうがんばっても30を超えられないのに，友人P氏は1回で気合98%。どうすりゃ気合を溜められるんだぁ？　(瀧)

◆絶対勝てると思って買ったチケットだったのに。指定席だったのに。畜生ぉー，ベルマーレぇー。チケットを買ったときと試合のときでこんなに変わるなぁー。おまけに大雨降るし……。でも，観戦に行くのはいいですよ。ストレス解消になります。ヤジで(と思うのはレッズファンだけ？)。

三浦　貴至(23)埼玉県

やはり負けてしまうとストレスが……。

◆“ふわふわ，もこもこ”とはチャチャが狼に変身したリーヤを形容した言葉です。ビー玉のような目がとってもらぶり～です。

津村　忠蔵(19)佐賀県

ということは狼を飼いたいということだったんですね。

◆この間，バイト先の電器店でお客様との会話。
老人「兄ちゃん，盗聴機あるけー！」
僕「え！」
老人「……あ，補聴器や！」
大声でやばいことをいわないでほしい……。

奥村　良行(23)三重県

そんなご老人には盗聴機がついた補聴器を売ってあげましょう(冗談)。

◆X68000ユーザーは実在した！(ドキュメンタリー番組じゃないって)。群馬大学の大講義室で，次のテストを受けるために座ってOh!Xを読んでいたら初代のユーザーに声をかけられてしまったのだ。私は答案を書いてすぐに退出したが，その後，彼はどうしているのだろう(できればまた会いたいものだ……つづく)。

戸谷　浩史(21)群馬県

続きを楽しみにしています。そのタイトルは「再会」とでも……(映画じゃないって)。

◆引越の準備もなんとか終わりつつあります。あと2～3日後が引越日です。コンピュータ関係も梱包してしまい，しばらくお休みです。引越後の開梱やセッティングでも，大変な毎日が続くと思います。忙しい時間の合間をぬい，このハガキを書いています。人間，忙しければ，忙しいほど時間の使い方が上手になるのは本当のようですね。あっはっはっは……。

壁谷　善嗣(35)宮城県

それでも時間の使い方が下手な人も……。

◆2年ぶり4度目なんていうと，まるで高校野球みたいですが，そうではなくて，キャンプに行きました。その日はよく晴れたのですが，実は1日中雨が降らなかったのは今回が初めてなのです。なにも私が雨男というわけではないと思うのですが，なぜか夕方または夜になると雨が降るのです。おかげで体育館とか車の中で寝ることに疑問を感じなくなりました。

藤原　彰人(24)岡山県

ということは，今回も車の中か体育館で寝たのでしょうか？　そんなわけはないか。

◆友人に「X68030がほしい」というと「今頃，なにをいっている」といわれます(周囲はPC-9801やMacintoshばかりなので……)。初めて買ったパソコンがシャープ製(MZ-2200)で，以来ずっとシャープ党だったので，他社のものに乗り換えることは考えられません。すると友人からは「マニアだなあ」といわれます。シャープのパソコンはマニアなものなのだろうか。うーむ。　湯舟　幸男(23)愛知県

しない後悔よりもする後悔ということで，友人の意見に惑わされず自分の好きなものを買うのがよいと思いますよ。

◆C言語を覚えるべく，努力している。以前，Oh!Xの読者に初心者はいないなどという発言が載っていたが，そのとき「俺って初心者だよなァ」と思っていた。プログラム言語のひとつでも覚えねば，中級者とはいえないと思っていたのだ。しかし，友人のひとりはX68000ACE-HDを持ってたが，今年になってハードディスクが内蔵されていることに気づき，フォーマットをかけてみると，どうやらクサッていたらしい。彼を見て，初めて俺って初心者じゃないんだと気づいた……。　廣瀬　研作(24)佐賀県

自分が必要だと思い，なにかを始めようと努力した時点で中級者といえるのではないでしょうか？　それを諦めるまでは……。

◆ドラマ「お金がない」が終了してしまった。自分自身が今年就職活動をしていたことや貧乏学生してることなどから，主人公の織田裕二にはとても感情移入してしまった。自分もこのドラマの主人公のように来年から「元気」をキーワードにして，“ビシッ”とやっていきたいと思っています。世の中，お金では買えないものがいろいろあるんじゃないでしょうか？

吉仲　正和(23)奈良県

ただ，あると便利なこともいろいろとありますし。たとえば，ハードディスクとかメモリとか……(アンケートハガキ参照)。

◆9月17日，近所の本屋へ行き「パソコン誌のOh!Xはもう出てますか？」と聞くと店員のおばさんがけげんそうな顔で「マゾコン(マザコンかも)誌の……ですか？」と聞き返してきた。ガクッときたが気をとり直して「パソコン誌ですが……」とやや強調して答えると「ああ！」とわかったような顔をして本棚のところへ行き，本を探し始めた。なんとそこは美少女ゲーム専門誌のコーナー……。もう間違いを正す気力もなく，しばらく待っていると「ないようですね」といってきた。「……そうですか」とだけ答えると本屋を出た。そして，次の本屋に向かうのであった(実話)。　原　弘樹(21)埼玉県

でも「マゾコン」や「マザコン」誌なんてあるんでしょうか？

◆10月号の清野一男さんへ。もう時期が遅すぎますので来年の春にでも，就職希望者のふりをしてどこかの会社にTELしてみてください。続けて，お知り合いの女性に同様のことをさせてみて，会社側の対応の違いを試してみてください。あなたの知人(男性)は本当に実力がなかったんだとわかるはずです。じゃね。

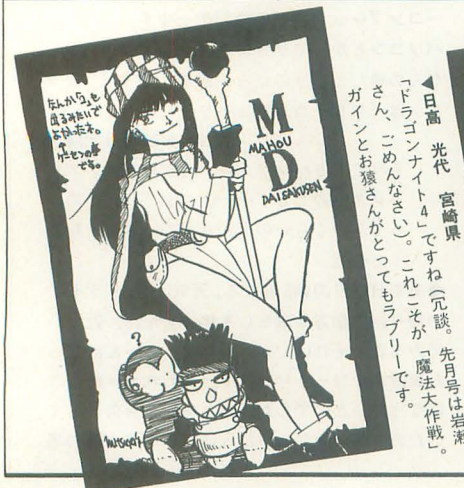
安藤　道子(22)宮崎県

真実かどうかわからないことを妙に納得するよりも，こんな方法で判断してみるのもいいかもしれませんね。ところで，年齢はこちらで推定させていただきました(1991年11月号参照)。

◆地区民運動会でセーラー服を着せられました(着せ替えレース)。スカートがすごく似合っていたので評判がよかったのですが，すね毛がちょっと……。　平　好博(23)富山県

いまごろはすね毛を剃って……。

◀日高　光代　宮崎県
「ドラゴンナイト4」ですね(冗談。先月号は岩瀬さん、ごめんなさい)。これこそが「魔法大作戦」。ガインとお猿さんがとってもラブリーです。

◀板橋　芳則　福島県
「銀の弾丸を撃ち込め」というタイトルだそうです。でも、銀の弾丸で倒すのって狼男だったような気が……。

◆寮で生活して，高校生のときにはゲームに熱中したけど，いまはそれほどではない人が多いことがわかってきた。逆に自分も含めたゲーセン好きも多少いることも，ゲーム機でちょっと遊ぶくらいならかなりの人数になることも。ゲームを4，5時間していても平気という人は，わりと少ないです。皆で騒ぎながらゲームをするのが好きなのです。さて，そんな寮生活の僕の部屋にはテレビがない！　同室の上級生がテレビ嫌いにより，この恐るべき生活に6カ月ほど耐えなければならない(他の部屋に行けばいいけど，夜中はさすがに行きづらいし……)。机はないは，ロうるさいは(ロぐせが「たとえばさぁ……」)。　小阪　友裕(20)石川県

確かに皆で騒いでいるんなら4,5時間でも平気かもしれませんね。ところで，これって同室の上級生は読んでいませんよね。読んでいたら，それも運命ってことであきらめてください。

◆瀧氏のTeX入門講座で「理系大学生は……」とありますがこれは，レポートを代わりにやってくれるものと解釈してよろしいっすね？……おい。　畑中　英喜(21)大阪府

本当に頼むんですか？　そんな無謀なこと私だったら怖くて……。

◆私にとっての一生の友人に結婚が決まった。そういうことには奥手と思っていただけに，驚きも喜びも大きい。打ち明けられた夜，彼と2時過ぎまで話し込んだ。いままでと変わらないことを照れくさそうに強調していた。過ぎゆく季節，変わりゆく人生に乾杯。

中島　民哉(24)埼玉県

変わりゆくものが多い世の中ですが，2人の友情だけは変わらないことを祈っています。

◆奨学金がもらえる身分になった……けど，自分が使うより先に，なぜか勝手に減っていく。親がローンの返済に使っているようだ。う〜む，どうするか？　太田　崇貴(23)岡山県

学費を親に頼っているならば，この現状は仕方がないという気も……。

◆ひと言：のりのつけ過ぎに注意！

尾澤　宏(22)兵庫県

ひと言：ごめんなさい。

◆ハードディスクを買おうと思っているのですが，僕のびんぼうゆすりに耐えられるかどうかが心配です。　片山　明義(16)奈良県

足に自信があれば，耐えられるかどうかお店で試してみましょう。

◆実に過ごしやすい季節になってきました。そろそろ卒業研究にかからなければと思うものの，気がつくとなにやら一生懸命makeしてたり……。「やっぱり楽しくないとね！」と思い，始めたものにズブズブはまっていく。まあ，卒業研究も楽しそうな内容だから……きっとはまれるでしょうね。少し眠い……。

中村　学(22)福岡県

卒論や卒研にはげまれている皆さん，追い込みがんばってくださいね。もちろん受験生も。

◀中川　和之　埼玉県
冬といえばやっぱり雪ですよね。背景は昨年冬を過ごした場所を参考にしたとか。イラストにあるような彼女と一緒だったのでしょうか？

◆それでは皆さん，よいお年を(これ，12月号のラストに使ってください)。でも同じこと考えてる人が8人くらいいたりして？

金子　直史(18)新潟県

どうやら，そんな人はいなかったようです。でも，今年中に1月号が出ちゃうし……。ということで，皆さんよい冬を。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

売ります

★ディスプレイテレビ「CZ-601D-GY」を送料込みで50,000円で売ります。箱，説明書はありませんが，付属品はあります。またシステムサコムのSCSIボード「SX-68SC」を10,000円以上で売ります。こちらは箱，説明書，付属品などすべてあります。連絡は往復ハガキでお願いします。〒194　東京都町田市藤の台団地2-10-102　青木　泰一郎(20)

★東京システムリサーチのメモリボード「XSIM M10」に10MバイトのSIMMを載せたものを45,000〜50,000円くらいで売ります。箱，説明書つきで送料込みです。値引き交渉には応じますが，高価買い取り者優先です。連絡は往復ハガキでお願いします。〒193　東京都八王子市横川町636-9　藤沢　実(20)

★X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を25,000円以下で売ります。送料込みで付属品はすべてあります。連絡は往復ハガキでお願いします。〒306-04　茨城県猿島郡境町1970-1　染谷達生(18)

★X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を20,000円で売ります。連絡は往復ハガキでお願いします。〒791　愛媛県松山市久万の台176-3 サンライズ第3マンション403　宮岡幸夫(18)

★カラーイメージユニット「CZ-6VT1」を30,000円(箱はありませんが付属品，説明書などはあり)で売ります。また，エプソンの熱転写カラープリンタ「AP1000」を25,000円(箱はありませんが，シートフィーダと説明書はあり)で売ります。アイ・オー・データ機器のX68000用2Mバイト増設RAMボード「PIO-6BE2-2M」を15,000円で売ります。連絡は往復ハガキでお願いします。〒651-22　兵庫県神戸市西区狩場台5-5-109　小西智也(41)

★システムサコムのSCSIハードディスク「HD-K520」(モッキンバード)を30,000円くらいで売ります。また，サンケンの無停電電源装置「MPS-500」を5,000円くらいで売ります。色はいずれも白です。連絡は往復ハガキでお願いします。〒408-03　山梨県北巨摩郡武川村上三吹345　輿石　学(24)

★X68000CompactXVI用2Mバイト増設RAMボード「CZ-6BE2D」を18,000円で売ります。箱，説明書，付属品などすべてあります。連絡は往復ハガキでお願いします。〒985　宮城県塩竈市芦畔町13-37　柳田正幸(23)

買います

★X68000用2Mバイト増設RAMボード「CZ-6BE2A」を25,000円(送料込み)で買います。箱はなくてもかまいませんが，説明書，付属品つきでお願いします。連絡は往復ハガキでお願いします。〒612　京都府京都市伏見区清水町883 ハイツシティ丹波橋10C　岩田友也(19)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は10月号の内容に関するレポートです。

●カラーページを開くと，SX-BASICによる美しいカードゲームやRPGっぽい画面，リストに直接貼り込まれたパターンが目に飛び込んでくる。「おお，SX-BASICもとうとうグラフィックを本格的にサポートして，こんなことが簡単にできるようになったのか，ブラボー！」と思ったのも束の間。本文を読むとリスト中のパターンはprint文用，RPG風サンプルは外部タスク，カードゲームは作者の努力と根性の産物……。「パワーアップの方向がちょっと違うのでは？」と思わざるをえませんでした。グラフィックサポートはGScriptだったし。RPG風タスクぐらい標準のSX-BASIC＆ウィンドウエンジンでできるようにしてもらいたいものです。よっぽど特殊なものでないかぎり，「やりたいことに合わせて外部タスクを作ってくれ」というのは明らかにまずいと思います。描画結果をGScriptで保存するのではなく，仮想画面にビットイメージとして保存するようなグラフィックアイテムを用意してもらいたいですね。そして，その領域に対して8ビット時代のBASICのようにprintとかputとかできるといいのですが。

あと，「TeX入門講座〜てふてふらてふ〜」はリスト＋詳しい解説のベストパターンで，さらに普通の文書を扱っているのもポイントが高いですね。TeXというとなんだか判読不能な変態文字列を，大量に打ち込む苦行の代償としてきれいな出力を得られる。というイメージがあって，それならGUIでWYSIWYGのほうがいいや，と思っていました。しかし，記事にある程度ならTeXを使ってもいいかな，と認識が変わりました。

石田　伯仁(21)　X68030，MZ-700，PC-9801VMII，PC-8801mkIIMR，PC-E200　神奈川県

●新製品紹介にあった「XL/Image」，なんか恐ろしいまでに本格的な仕様ですね。雰囲気としては「3Dレンダリングシステム界のGCC」といったところでしょうか。ワークステーションの天下り(?)は大歓迎したいです。これまで発表されてきたレンダリングシステムは，まず「処理速度の遅さ」ゆえにユーザーを自ら遠ざけてしまった感じがします。実際のところ「3DCGってのは日数のかかるもんや」といった，職人気質をもった人しか使っていないのではないでしょうか。現在，X680x0で3DCGしている人のほとんどは「DōGA CGAシステム」を活用しているのではないでしょうか(勝手な憶測ですが)。そのシステムと同等の処理速度でレイトレーシング画像ができるとなると，ひょっとして「MATIER」が登場したときと同じ衝撃が，3DCG野郎にも走るかもしれませんね。いままでの「3DCGならDōGA CGAシステム」というイメージを変えるほどのソフトであってほしいですね。

あと，どうせかうなら「X」だよな，と思いつつ買ってしまったX68000版「スーパーストリートファイターII」ですが，「餓狼伝説SPECIAL」以上の移植完成度。本当にカプコンはX68000によくやってくれました。でも，へたすりゃこの「スーパーストII」でX68000はおしまいにするんじゃないかって，思ったりもしています。もしも，次回作が「ヴァンパイア」なんてものが予定されていたらすごいんですけど。個人的にはぜひ移植してもらいたいぞ。

中矢　史朗(23)　X68030/X68000 ACE-HD，PC-386P　愛媛県

●付録ディスクに収録されていたXSPRITE.FNCはコケると思います。Oh!Xでは，この手のものの発表が多いけれど，どうなることやら。しかし，気持ちはわからないこともありません。ゲームを創造する力がなければ，どうしてもゲームを作ることができないものです。そういう人はこういったものを作ることに走ってしまうのもしかたがないことです。確かに「ゲームのアイデアがどうしても出てこない」という人はたくさんいるでしょう。それもそのはず，創造するためには創造する訓練というものが必要なのです。それと正しい考えと敬虔な気持ちも必要です。ゲームを作ろうとしないで，市販のゲームばかり遊んでいるという片寄った生活をしていると，「ゲームは本当に自由に作れるんだ」ということを，あまりにあっさり忘れてしまいます。オリジナルのゲームはまだまだあります。自由な発想でプログラムを作ろうじゃないですか。

鈴木　朝夫(20)　X68000，X1 turboZ，MZ-731，PC-9801RA，PC-8801VA，PC-6001mkII，FM-77AV，MSXturboR，ZX81　神奈川県

●最近付録ディスクがついてきますが，初期のものに比べて，内容が片寄っているのではないでしょうか。確かにディスクで配布されるツールの1つひとつが，かなり大規模なものもあり，磁性面的にもきついことがあるでしょう。また，現在ではX68000のDOS環境における一般的なツールがかなり揃っています。そういう意味では，「すべての人に役立つツール」はそう簡単に作れそうもありませんが，やはりなにかひと味足りないような気がします。

しかし，「SX基本ツール」と「SADJUST.R」は非常に役立っています。地味ながらも「2行にするの.R」は重宝してますし，「SADJUST.R」も必要としている人が多いでしょう。細かい設定もできて便利ですね。

奥田　直也(22)　X68030/X68000 ACE-HD/SUPER，MSX2，PC-E550　愛知県

ごめんなさいのコーナー

10月号　特別企画「もみじ狩りPRO-68K」

X-BASIC用外部関数「XSPRITE.FNC」と「EXEC.FNC」で，同一名の外部関数が存在したためサンプルが正常に動かない場合があります。両関数を使用する場合は，X-BASICへ同時に組み込まないようにしてください。

12月号　特別企画「懐ゲー制作工房」

P.52　逆襲のバニーガール

ゲームは正常に遊べますが，メッセージ部分で間違いが見つかりました。BUNNY.Hの12行目の「〜，"ドライブしな"，〜」を「〜"ドライブし"，〜」に訂正してください。

P.61　世紀末大戦術

都市を全部守り切ったときと難易度調整に不都合がありました。58ページにある「世紀末大戦術のデバッグ」のカコミに従って，付録ディスクから解凍したソースリストに訂正を加えてください。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

誌面がワカメになっていないといいなあ

▼またもや付録ディスクがやってきました。今回は，国内最強のレンダラといわれる「XL/Image」のお試し版といろいろなアプリケーションを収録しました。お試し版ということで，使える機能は限られていますが，「XL/Image」の噂の表現力を自分の手で確認してみてください。

また，「こちらシステムX探偵事務所」で制作されていたピンボールゲームも収録されています。目指せジャックポット！　を合言葉に遊び倒しましょう。

「XL/Image」については，製品版をもとに引き続き活用法をレポートしていく予定です。お楽しみに。

▼そして，この12月号をもって，Oh!Xが7周年を迎えました。今回は特別企画「懐ゲー制作工房」と題して，いくつかの懐かしい雰囲気をもつゲームたちを再現してみました。記事を読んでいただければわかるとおり，久しぶりにリストだらけの特別企画となっています。

最近では，雑誌にディスクがつくのが当たり前のようになってしまいましたが，ほんの数年前までは，このような大きなリストたちがあちこちの雑誌で見られたものです。

しかし安心してください。ソースリストはちゃんと付録ディスクに収録されています。まあ，昔ながらの体験をしたいというならば，自分でリストを打ち込むことは止めませんが……。それはともかく，付録ディスクに収録されているのはソースリストのみです。特別企画の記事をよく読み，必要に応じてコンパイル，アセンブルしてください。

なかにはマスターアップに間に合わず，訂正が必要なものもありますので，こちらも特別企画の記事，ごめんなさいのコーナーに従って訂正しておいてください。

▼「ハードコア3DエクスタシーSIDE A」は著者多忙のため，「X68000マシン語プログラミング」は，先月同様お休みとなってしまいました。ごめんなさい。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶冬コミで同人誌ほかを出す予定。内容はゲーム攻略もの。メインディッシュは，「VIEW POINT」のハーデストのノーミスクリア完全攻略。限定20本でクリアビデオも出す予定。PCエンジンの「ときめきメモリアル」の攻略も掲載するか？　そうそう，チェルノブアダプタも100個ぐらい生産する予定。詳しくはまた来月の編集後記で。（瀧）

▶しかしMarsTVが1カ月で終わるとは。せめて3カ月はもてよ（笑）。あぁ，これでもう吉村君に出会うことはないだろう。合掌。話はさておき最近はピッカピカ天素の銀座7丁目新人コーナーが面白い。やっぱり必死で無茶やってる奴が強いなあ，と実感してしまうのでした。無茶やりすぎて死人出しちゃうO川興業はちと困ったちゃんだけど。（哲）

▼締め切り直前にいきなりMOクラッシュ。失意の中，石田氏のピコピコエンジンを見せられ，執念の復帰。皆さん，MOのバックアップはこまめに取りましょう。それから，今回のSX-BASICでHから始まる変数名が使えないことが判明しました。この症状がどうしても困るという方は，comp.cの中のstateHの中身をreturn(FALSE);に変えてください。（石）

▼ビートたけしが事故を起こしてからずいぶんたち，もう退院すらしたというのに，いまだに元気な姿をテレビで目にすることがたまにある。「事前に収録したものです」というテロップが出るものの，かなり撮り溜めしてたことがばれちゃったよな。そういう番組って，あまりよいイメージを受けないと思うのは私だけだろうか。（I.K）

▼先日テレビでサザエさん25周年記念スペシャルを見た。サザエさんの25年後の様子というのが衝撃的。タラちゃんがビール飲むし，カツオはハゲてるし。それと20数年前に放映された初期のサザエさんも公開していたが，フネが妙に暴力的で恐怖を覚えた。それにしても私は笑う声もサザエさんと同じだがサザエさんではない。ないといったらない。（善）

▼縁あってPlayStationの仕事をすることになった。いま一般にも3DCGが認知されようとしている。もはや表示できればよいという段階は過ぎた。表現の方法論を確立すべき時期なのだ（その意味で車モノは楽な部類）。本誌での連載をそのための理論構築の場としていければと考えている。というわけで今月はとりあえず休載です。ごめんなさい。（A.T.）

▼盛り上げるだけ盛り上げといて，最後は巨人と西武だなんて，できすぎてると思ったのは私だけではあるまい。サントリーシリーズでメロメロなのにニコスシリーズではちゃっかり優勝してついでにチャンピオンシップも勝って，ってヴェルディのシナリオも2年連続になりそうな気がしてきた。ブランド力とはそういうものなのかもしれない。（K）

▼ジャストからようやくアクセラレータが発売された（ただ発売後の連絡がないので……）。ただ，スピードに関しては想像どおりという感じだが，別の会社からもアクセラレータが発売される予定。ローカルメモリも積んでるし，値段はともかくスピードは結構期待できるかも。DSPボードのほうもそろそろみたいだし。まだまだ楽しみは尽きないかな。（高）

▼人間がいい加減なせいで，記憶力には自信がない。頭の悪さに加えて，はなから正確さを重視していないからどうしようもない。ニュアンスさえ把握していればいいと思っているので，記憶を探るのもあいまい検索だ。アバウトな性格はあまり不便も感じないけど，カラオケで思わぬ曲が出現するのはちょっと困るかも。タイトルと曲がいつも違う……。（ふ）

▼ただいま「カービィボール」に夢中。とりあえずエキストラコースは遊べたし，サウンドテストメニューも見れた。あとは，エキストラコース全部で銀メダルを取ることが目標かな。もしも読者のなかで，エキストラコース全部で銀メダルを取るとなにが起こるか知っている人がいても，絶対に教えないでね。楽しみがなくなっちゃう。（J）

▼先日，「シャープに爆弾を仕掛ける会」の会長だった人と次世代ゲーム機について長電話した。CPU性能を初めとしたスペックだけは凄いが，周りが追い着いていないので，どう見ても，やっぱりおもちゃには違いない。可能性も限界もなんとなくわかってきたところで，そろそろ次々世代機というものが見え始めるのかな。（U）

▶「1年間で結果を出す」といったのは次期日本代表監督に内定の加茂周監督だが，シャープにもそういう態度というか決意を見せてもらいたい。デザインだけでも発表するぐらいのことは必要だと思う。はっきりいうとユーザーもOh!Xもいまのx68000/30でそれほど困っているわけじゃない。不安なのは将来の身の振り方だからね。（T）

microOdyssey

1日の周期は24時間ということになっている。これは地球の自転の関係というか，日が昇って沈み，再び昇るまでの時間がだいたい24時間だからであろう。まあ，高緯度地方のことは考えないことにしよう。

私事で恐縮だが，こんな生活をしていると朝を会社で迎えることが多い。そして，1日という単位が24時間という決まったサイクルではないのだ。ふと思うのだが，人にとっての1日は本当に24時間なのだろうか？

こんな実験があったそうだ。それは，人に昼か夜か時間がまったくわからないような場所で生活してもらう。そうすると，個人により周期に多少の違いはあるものの，約25時間を1日として生活するらしい。

ということは，人が好きなように生活していくと1日につき1時間くらいずれていく勘定になる。おお，これでは日常生活に支障をきたすはずである。にもかかわらず，社会は破綻せずに(本当はすでに破綻しているのかもしれないが)機能している。これは人の義務感や責任感のなせる業なのだろうか。たとえば，会社に行かないと給料がもらえないし(行っただけではもらえないが)，授業に出席しないと単位が取れないから，25時間周期を崩してまでも朝早く起きるのかもしれない。

さて，1時間のずれはなぜ問題になってこないのだろうか。これには光が大きな影響を与えている。つまり，朝がくるから人は24時間周期で行動できるのだ。というのは，眠っているときにある一定時間の光を当てると周期がずれて早く起きる(この場合は，24時間周期に同調する)らしいのだ。

だから，人は24時間周期の生活が送れる……ただし，これは朝起きて夜に眠る人の場合だ。これでは私の状況にはあてはまらない。

一般的に時間のずれということでピンとくるのが海外などに行ったときに起こる時差ボケ。たとえば，日本からヨーロッパに行く場合，日本で眠る時間を約8時間遅らせた感覚である。また，アメリカに行く場合は約7時間早く眠るような感覚になる。1日の周期が24時間より長いことを考えれば，後者のほうがつらい。

また，夜間や早朝に働いている人はどうだろう。朝から昼にかけて眠ることになれば，明るさや騒音のために眠りを妨害されることが多い。早朝に働く場合，前日に早く眠ろうとするだろうが，25時間の周期を考えるとそう簡単には眠れない。はたして，交代勤務などで不規則な生活を続けているとまともな周期に戻すことさえ困難になるかもしれない。

25時間周期ということを考えると，人はもともと夜更かしをしやすい生き物なのかもしれない。それが，科学の発達により，夜もまるで昼間のように明るくなり，夜更かしを助けるようになってきた。自分の身体のことを考えるなら，規則正しい生活，それが無理ならある一定のリズムで生活を送りたいものだ。

しかし，1日の周期が25時間になれば，1カ月にして約30時間，いまの感覚でいうと1日ちょっと締め切りが延びることになるのか……いいかもしれない。でも，それっていうのは結局，働く時間がさらに増えるってことか。24時間周期のままでいいかも。　(高)

1995年1月号12月17日(土)発売

特集　割り切って使うCD-ROM

・各種ドライブ紹介　・他機種CD-ROMの利用

新製品紹介　Xellent30

THE SOFTOUCH　魔法大作戦/VIEW POINT

全機種共通システム
シューティングゲーム作成講座(5)

バックナンバー常備店

地域	地区	店名・電話
東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートB1 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3209)0656
	渋谷	大盛堂書店 03(3463)0511
	池袋	旭屋書店池袋店 03(3986)0311
	八王子	くまざわ書店八王子本店 0426(25)1201
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 043(224)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

12月号
■1994年12月1日発行　特別定価900円(本体874円)
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印　刷　凸版印刷株式会社

満開の電子ちゃん

作・え 岡村祭

78号(10/18発売)は、光通信でザウルス⇔X68のデータをやりとりしたり、IVM解析資料が載ったり。スーパーストIIのパッチもあるよん。もちろん2枚組だ!!

谷口 博一(京都府)

私も「電クラ」の購読を始めてはや四年目を迎えました。定期購読がこんなに続いた雑誌も初めてですが(Oh!Xがあるけど)、なんでこんなに続くのでしょう? それは「電源オンですぐ起動、マウスひとつでらくらく操作」だからです。内容の充実はもちろんのこと、編集者の方々の並々ならぬ苦労、パワフルな読者の参加も大変なものです。

私もこんなかたちで「電クラ」に参加していますが、皆さんも購読を始められて参加されてはいかがですか? ウハウハですよ!

恒例「シャープわんさかバザール」開催!!

期間：11月19日(土)～11月30日(水)まで

シャープフロアが本店4Fに移動!!ますます充実して新装オープン!!

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO

お申し込みは今すぐ！ 受注専門フリーダイヤル **0120-377-999**

場所：ツクモパソコン本店4F

期間中、パソコン本体をお買い上げの方には今回特製オリジナルグッズ「X68030電飾POPスタンド」をプレゼント！←先着50名様まで

特価品はお店に来ないとわからない!!急いで!!

X680x0シリーズ

本体

CZ-674CH(X68000 CompactXVI) 超特価 **¥82,800**

TS-XFDCAを使えば、縦置き5インチモデルX68000シリーズ(PROシリーズを除く)を外部ドライブとして使用可能！

是非、2台目のマシンとしてどうぞ！

※モニタ別売です

お勧めのセット!!

CZ-674C-H････ ¥298,000
CZ-608D-BK･･ ¥ 94,800

ツクモ特価¥146,800

X68030 お勧めの組み合わせ!!

CZ-500C-B････ ¥398,000
290MBハードディスクサービス

ツクモ特価¥280,000

※モニタ別売です

満開製作所の商品も取扱中！

X68000 CompactXVI 24MHz改

商品	価格
RED ZONE	ツクモ特価¥ 98,000
RED ZONE(2DD)	ツクモ特価¥103,000

満開製外付け5インチFDD

商品	価格
MK-FD1	ツクモ特価¥ 39,800
MK-FD1(カラーリングモデル)	ツクモ特価¥ 44,800

X680x0シリーズ用RAMボード

商品	対応	ツクモ特価
SH-6BE1-1ME	(CZ-600C専用)	¥10,500
PIO-6BE1-AE	(ACE/PRO/PRO2シリーズ用)	¥10,500
PIO-6BE2-2ME	(拡張スロット用)	¥22,500
PIO-6BE4-4ME	(拡張スロット用)	¥38,200
SH-5BE4-8M	(X68030シリーズ用)	¥44,000
CZ-6BE2A	(XVI専用)	¥42,500
CZ-6BE2D	(CompactXVI専用)	¥29,800
TS-6BE2B 在庫限り	(CZ-6BE2A/D用拡張RAM)	¥29,800
X SIMM10-8M	(拡張スロット用8MB)	¥53,800
X SIMM10-10M	(拡張スロット用10MB)	¥64,800

X680x0ユーザーの為のツクモオリジナルシリーズ

TS-3XRシリーズ　X680x0用3.5インチ外付けドライブ

●2DD/2HD/2HC/1.44MBフォーマット対応
※2DD/2HC/1.44MBを使用するにはHuman68K Ver.3.0以上が必要
●CompactXVI/68030用ケーブル付

TS-3XR1B 1ドライブ 定価¥33,800……**ツクモ特価¥26,800**
TS-3XR2B 2ドライブ 定価¥46,800……**ツクモ特価¥36,800**

簡単コンピュータミュージック

Music Card for X680x0 (TS-6GM1)

●音源を搭載したMIDIボードの登場。これ1枚で手軽にMIDIコンピュータミュージックが楽しめます。GM規格・MT－32・CM－64等の音色配列をサポート。最大同時発音数16。

「Mu－1GSお試し版」付き

ツクモ特価¥24,800

ジョイスティックパラレルインターフェイス

●拡張スロットを使用しません。ジョイスティック端子に接続できるパラレルインターフェイスです。これでスキャナーも高速で取り込みが可能になります。
★取り込みソフトェア及びサンプルソース付属。

TS-JPIFS(CZ-8NS1対応用)……定価¥17,800……**ツクモ特価¥14,800**
TS-JPIFE(EPSONスキャナー対応用) 定価¥17,800……**ツクモ特価¥14,800**

ツクモオリジナルバージョン・X68030 HG

X68030 (CZ-500C)

★内蔵500MBハードディスク
★8MB増設メモリーコプロセッサ
★SX WINDOW V.3.0 プレインストール済み
以上全てを内蔵済みで・・・

ツクモ特価¥368,030

※ニューセンター店のみのお取り扱いです。

CZ-500C-B用

500MB内蔵ハードディスク

ツクモ特価¥68,030

※ニューセンター店にてお取り付け致します。

オリジナルSCSI&RAMボード

NEW **TS-6BS1mkII** 近日発売予定!!

変更点 その1　接続コネクタをフルピッチからハーフピッチコネクタに変更致しました。

変更点 その2　72PINのSIMMメモリソケットを、一つ用意しました。これにより拡張スロット不足でお悩みの方に朗報です。

試作品

ツクモ特価¥39,800

X68000Compact/RED ZONE用内蔵6MB+FPUボード

TS-6BE6DP

※FPUにMC68882を使用しているため、HumanVer3.0より前に付属していたFLOAT3.Xでは使用できませんのでご注意ください。

定価¥64,800 **特価¥57,800**

プリンター

マッハジェットカラー
MJ-700V2C(ケーブルセット)
ツクモ特価 **¥71,000**

バブルジェットプリンター
BJ-10VLite(ケーブルセット)
ツクモ特価 **¥29,800**

カラーバブルジェットプリンター

BJC-400J(ケーブルセット)
モノクロ高速
カラー対応
エコノミータイプ
ツクモ特価 **¥61,000**

BJC-600J(ケーブルセット)
カラー高速印字
スタンダードタイプ
ツクモ特価 **¥71,000**

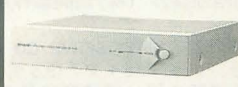
ディスプレイも特別価格にて提供中！

CZ-608D(14型カラーディスプレイ) ツクモ特価¥ 66,000
CZ-615D(15型カラーディスプレイテレビ) ツクモ特価¥132,000
CZ-621D(21型カラーディスプレイ) ツクモ特価¥125,000

映像関連機器

動画を始めてみませんか？

ビデオ入力ユニット
CZ-6VS1 定価¥178,000

MC68EC020(25MHz)の32BitMPUを搭載し、SCSIインターフェイスを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。但しX680x0シリーズでご使用の場合には6万5千色までの表示となります。

ツクモ特価 **¥142,000**

多機能対応型スキャンコンバーター

電波新聞社**XVGA-1v**
定価¥69,800

X680x0シリーズやその他のパソコンの水平周波数(24KHz/31KHz)をNTSC標準信号に変換するスキャンコンバータユニットですので、家庭用テレビやビデオ・デッキで映像を表示または録画することができます。また、ビデオプリンターを使えば画面のハードコピーも可能です。

ツクモ特価 **¥59,300**

ビデオプリンター(昇華型)

シャープ **VP-ES1**

NEW

高画質ハガキ大プリント、普通紙・布転写用紙もOK。4分割、16分割、ストロボも可。※入力信号は、ビデオ信号となりますので、パソコンに接続の場合にはお問い合わせ下さい。

ツクモ特価 **¥49,300**

【東　京】●パソコン本店（各種パソコン・周辺機器）●パソコン本店II（パソコン・ワープロ）●DOS／Vパソコン館（DOS／Vパソコン・下取り）●万世店（総合通信機器）●5号店（ビデオ・ムービー・CS）●ソフト8号店（パソコン&ゲーム用ソフト）●買取センター（ゲーム機・ゲーム機用ソフト買取り）●ニューセンター店(パソコン・中古・下取り・買取り)●ラジオセンター店（ハンディーレシーバー・テレホンパーツ）
【名古屋】●名古屋1号店（パソコン全般）●名古屋2号店（パソコン全般・総合通信機器・ビデオ）【札　幌】●札幌店（パソコン全般・総合通信機器）●DEPOツクモ札幌（パソコン全般）

THE AKIHABARA EXCITING FESTA

秋葉原電気まつり:11月19日(土)～'95/1月8日(日)

5千円お買い上げ毎に抽選券1枚プレゼント。1等:10万円　2等:5万円　3等:5千円 賞金総額6,000万円

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO

受付時間(平日)AM10:45～PM7:30
(日・祝) AM10:15～PM7:00　木曜定休

『FAX24時間お見積もり受付』 03-3255-4199
お名前.住所.電話番号.FAX番号をご記入の上ご依頼下さい。

ツクモグローバルJCBカード

JCBならではの国内・海外サービスにツクモオリジナルの特典をプラス。ツクモ各店にある入会申込書にてお申し込み下さい。くわしくはグローバル事務局03(3251)9898又は各店へ。
※ジャックス・VISA・セントラル・マスターも取り扱っております。

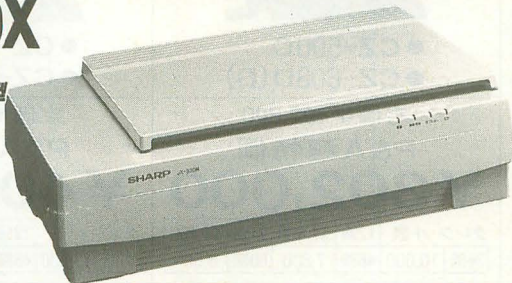
カラーイメージスキャナー

JX-330X

定価¥178,000

ADF・透過原稿対応型カラーイメージスキャナの登場です。

高解像度(600dpi)、超高速が特長です。

ScannerTools(画像入力ソフト)付属。

ツクモ特価 ¥138,000

台数限定 **CZ-8NS1……ツクモ特価¥69,800**

コンピュータアート スーパーグラフィックツール

その1 慣れてしまうとマウスがいらない

DrawingSlate	¥74,800
Matier Ver2.0	¥39,800
合計定価	¥114,600

ツクモ特価¥ 85,000

その2 ハイクオリティなのにこんなに安い

MJ-700V2C	¥99,800
プリンターケーブル	¥ 4,800
Matier Ver2.0	¥39,800
合計定価	¥114,400

ツクモ特価¥106,000

大容量記憶装置

MO特選セット

SCSIボードが必要な場合にはセット価格に¥22,000加算となります。

機種	セット内容	ツクモ特価
Logitec LMO-400 (230MB) ¥158,000	プラス MOメディア SCSIケーブル ターミネータ	¥112,000
Logitec LMO-200 (128MB) ¥79,800	プラス MOメディア SCSIケーブル ターミネータ	¥ 69,000
Panasonic LF-3200JD (230MB) ¥168,000	プラス MOメディア SCSIケーブル	¥128,000
ELECOM EMO-L230 (230MB) ¥128,000	プラス MOメディア SCSIケーブル	¥128,000

ハードディスク

290MBハードディスク	ツクモ特価¥ 29,800～
350MBハードディスク	ツクモ特価¥ 44,800～
500MBハードディスク	ツクモ特価¥ 49,800～

CD-ROMドライブ

メーカー	機種		ツクモ特価
ナカミチ	MBR-7	(トレー)	¥44,800
メルコ	CDS-E	(SONYトレー)	¥24,800
Logitec	SCD-400	(NEC製4倍速)	¥37,800
PIONEER	DR-U104X	(4倍速)	¥64,800
PIONEER	DRM-602X	(6連装2倍速)	¥55,800
AIWA	ACD-300WN		¥26,800

CD-ROMドライバーソフト+SCSIケーブル ¥ 9,000

MIDIコンピュータミュージック特選セット

Roland SC-55mkII セット

SC-55mkII	¥69,000
TS-6GM1	¥39,800
MIDI変換ケーブル	¥ 4,000
合計定価	¥112,800

ツクモ特価¥79,800

Roland SC-55mkII 互換セット

MC6600	¥49,800
TS-6GM1	¥39,800
専用MIDIケーブル	¥ 3,500
合計定価	¥93,100

ツクモ特価¥64,800

Roland SC-88 セット

SC-88	¥89,800
TS-6GM1	¥39,800
MIDI変換ケーブル	¥ 4,000
合計定価	¥133,600

ツクモ特価¥99,000

電子文具

これぞ、パーソナルシステムの決定版!! Ink ZAURUS (PI-4000シリーズ) 登場!!

PI-4000 定価¥75,000 ツクモ特価¥59,800

PI-4000FX 定価¥91,000 FAXモデムセットモデル ツクモ特価¥72,800

※パソコンリンクソフト SX-WINDOW デスクアクセサリ集 ツクモ特価¥11,800
※パソコンリンク用ケーブル CE-150TS ツクモ特価¥10,200

パソコン通信

モデム

US Robotics Sportster 28800FAX 特価¥38,000
US Robotics COURIER V.34 TERBO 特価¥63,800

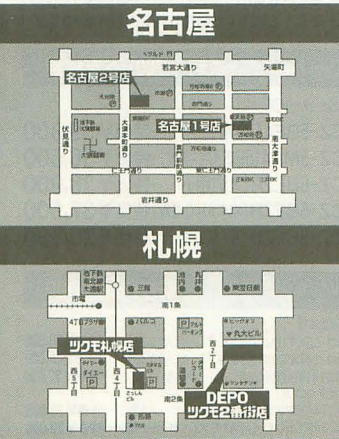
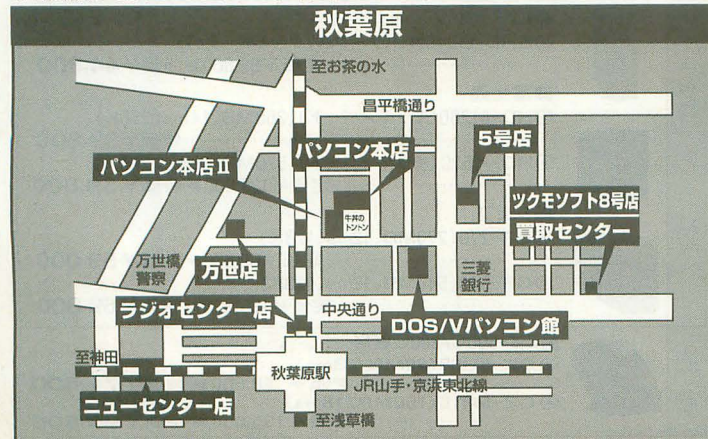
		ツクモ特価
AIWA	PV-BF144	¥17,800
OMRON	ME1414B II (NEW)	¥19,800

通信ソフト

		ツクモ特価
SPS	た～みのる2	¥13,000
SHARP	Communication SX-68K	¥15,800

ソフトウェア

ソフト	ツクモ特価
OS-9/X68030 V2.4.5	¥20,000
Technical Tool Kit V.2.4.5	¥16,000
Ultra C & Professional Pack V1.1	¥36,000
X Windows V11.5	¥24,000
SX-WINDOW Ver3.1システムキット	¥18,200
SX-WINDOWデスクアクセサリ集	¥11,800
C COMPILER Ver2.1 NEWKIT	¥35,800
Easydraw SX-68K	¥15,800
Easypaint SX-68K	¥10,200
SOUND SX-68K	¥12,600
Communication SX-68K	¥15,800
Matier Ver2.0	¥29,800
CD-ROM Driver	¥ 4,320
SX-PhotoGallery	¥14,220
DoubleBookin'	¥11,520
SX広辞苑(CD-ROM別)	¥17,800
EGWord SX-68K	¥47,800
SX-WINDOW開発キット	¥31,800
開発キット用ツール集	¥10,200
倉庫番リベンジSX-68K	¥ 5,400
MUSIC SX-68K	¥30,400
XDTP SX-68K	¥28,000
フォント&ロゴデザインツール書家万流SX-68k	11月発売予定
Super BUSINESS	NOW WAITING

★広告掲載価格は変動することもございます。お問い合わせください。

お支払い方法

あなたのご都合に合わせていろいろ選べます。

クレジット払い

月々¥3,000以上の均等払いも頭金なし。夏・冬ボーナス2回払いもOK!

カード払い

¥5,000以上
通信販売での御利用カード
ツクモグローバルカード・セントラル・ジャックス
※御本人様より電話で通信販売部へお申し込み下さい。

各種リース払い

詳しくは各店にご相談下さい。

現金書留払い

〒101-91 東京都千代田区神田郵便局私書箱135号
ツクモ通販センター Oh!X係

代金引き換え配達

お申し込みは電話1本でOK!
配達日の指定もできます。

銀行振込払い

事前にTELで振り込み先をご連絡下さい。
三和銀行 秋葉原支店
(普) 1009939 ツクモデンキ

★12月はツクモ全店休まず営業致します。

商品についてのお問い合わせは各店に

秋葉原

(営)平日AM10:45～PM7:30・日・祝AM10:15～PM7:00

ツクモパソコン本店 4F
03-3253-1899
03-3253-5599(代)
(休)木曜日

ツクモニューセンター店
03-3251-0987
(休)木曜日

名古屋

(営)AM10:00～PM7:00
(12月よりAM10:30～PM7:30)

ツクモ名古屋1号店
052-263-1655
(休)火曜日

ツクモ名古屋2号店
052-251-3399
(休)水曜日

札幌

(営)AM10:30～PM7:30

ツクモ札幌店
011-241-2299
(休)木曜日

(営)平日AM10:40～PM7:30 日・祝AM10:10～PM7:00

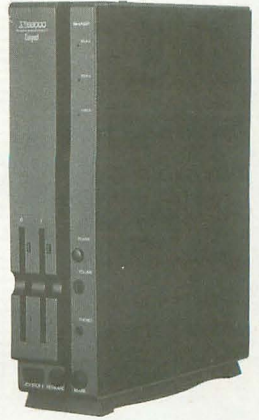
DEPOツクモ2番街店
011-242-3199
(休)木曜日

★商品はお電話受け付けより、標準日数3日～1週間でお届け致します。(一部地域を除く)
★表示価格には消費税は含まれておりません。

安いのに親切
TSUKUMO

九十九電機株式会社

RED ZONE 98,000円に大幅値下げ！
SCSI-2の2.4倍速CD-ROMでウハウハだ！

品名	価格
・RED ZONE（コンパクトXVI24MHz改造機）	¥98,000
・RED ZONE（FDDを2DD改造）	¥103,000
・マウスジャック（X68で98用バスマウスを使用する変換ケーブル）	¥4,000
・MK-FD1（オートイジェクト5インチFDD2ドライブ）	¥39,800
・メルコ製CD-ROMドライブCDS-E（2.4倍速＆計測技研ドライバ＆ユーティリティ付属）	¥29,800
・MK-HD1-EX（1GバイトHDDミニタワースロット4基）	¥145,000
・MK-HD2（1GバイトHDDユニット）	¥125,000
・X68030D'ash（33MHz＆コプロ内蔵＆MPUをMMU付MC68030に交換）	¥350,000
・コプロセッサ68882RC25（XVI・REDZONE用）	¥12,000
・コプロセッサ68882FN25A（X68030用）	¥10,000

RED ZONEがさらに
お買い求め易くなりました！

直販でのご購入の方法

当ショップは通販専門です。電話・郵便・ニフティサーブ（GO MANKAI）でご注文を承っております。
お支払は、代金引換（着払）、カード払い（合計9,000円以上、JCB、VISA）、郵便振替、現金書留（下記住所宛）のいずれかをお選びいただけます。
郵便振替口座：00150-3-568201口座名パソコンショップ満開
即納可能でない商品もございますので納期はお問い合わせ下さい。

（株）満開製作所 **パソコンショップ満開**

※価格には消費税は含まれておりません。

〒171 東京都豊島区長崎1-28-23 Muse西池袋2F
TEL.03-3554-7441

初雪や、こたつとみかんとディスプレイ、ジャストのX68kペリフェラル

いやー、みなさんご心配おかけしました。懲りずに落とし、また懲りずに復活（失笑）、ジャストの広告は永遠に不滅です。・・・わかってます、みなさんごめんなさい。もうしません。だから、石とか386とか投げないでくださいね。

●MPUアクセラレーター *H.A.R.P* for MC68000

型番：DCMA00D1　対応機種：X68000初代、ACE、EXPERT、PRO、SUPER　定価￥29,800（税別）

強制水冷でクロックスピード100MHzオーバーなどと、浮世の騒ぎも世紀末といった状況の今日この頃、X68kユーザーの皆様、いかがお過ごしでしょうか？（石はダメよ）。次期主力プロセッサーを争う開発競争も一段と激しさを増す中で、時代に取り残されてしまうのかM68000アーキテクチャー！などと大騒ぎするほどの事ではありませんが、いま業界で「静かな（笑）」トレンドとして注目されつつあるM68000アーキテクチャーでのアクセラレータMPU、そのエントリークラスとして開発されたH.A.R.P. for MC68000が、あなたとあなたのX68kの強い味方になります。ソフトウェアの互換性を維持しつつ、プロセッサーの演算速度を倍速化。危なげな改造をすることなく、20MHzドライブがMPUチップを差し替えるだけの簡単な作業で取り付け完了です。さらに、弊社製SIMMボードのER10と組み合わせれば、メモリーアクセスの高速化さえ実現します。H.A.R.P. for MC68000、いかがすかぁ。

●拡張SIMMメモリーボード *ER10S*

型番：ER10S0n（SIMM未実装）　対応機種：X680x0全機種
定価￥14,800／ER10SDn（4MByte SIMM1枚実装済）定価￥39,800（定価はすべて税別）

さて、前述のとおり（何が？）の状況で、安くなってきました72ピンSIMM。AT対98の抗争は、いつも、poorな私達に思わぬ福音をもたらしてくれています。その安いSIMMをより手軽に、というわけで、やっぱりER10Sしかないという手前味噌な話です（笑）。
MPUの演算速度が上がっても、バスやI/OがボトルネックとなってはせっかくのMPUアクセラレーターがもったいない、との考えから開発されたメモリーボードことER10S、搭載するメモリーにはローコストかつ高速なアクセスタイムを持つ製品が容易に手に入る72ピンタイプのSIMMが利用できる設計とし、さらに内部で使用するゲートICなどにも高速な製品を採用して、いかがわしい改造を施されてしまったX68kにも余裕を持って対応できると確信しています（具体的な保証はできませんが＾＾；）。さらにH.A.R.P. for MC68000との組み合わせによって、ER10S独自のメモリーサイクルアーキテクチャーによるメモリーサイクルの短縮を実現、標準4クロックで行われる。メモリーサイクルを3クロックとして、バスのボトルネックさえ克服します。と、偉そうなことをいろいろご説明しましたが、まあ、ER10S、買ってください。お願いします、って話です。実に単純明解ですね。

●MPUアクセラレーター *H.A.R.P-FX*（H.A.R.P for MC68030）

型番：DCMA30F1　対応機種：X68030をはじめ、MC68030（PGAソケット）が採用されコンピューターシステム（供給クロック25MHz以下）
定価￥68,030（税別）

X68030をはじめ、MC68030が入っているコンピューターをお持ちの方、注目です。PGAタイプMPUピン互換による2倍速MPUアクセラレーターことH.A.R.P-FX。50MHzの実力は、伊達じゃないです。乞御期待。

●拡張I／Oスロット *ESX68*

型番：ESX68L4　対応機種：X680x0全機種　定価￥39,800（税別）

拡張スロットが足りない、もちろん経済力だって、っと開き直れる話ではありませんが、少なくとも前者は如何ともしがたいものがありますね。そこで、ESX68があなたのお役に立ちにます。ひとつ、よろしくです。

今年、これだけはないと思っていた巨人の優勝、信じて疑わなかったヤクルトおばさんのスワローズ優勝記念セールは、神宮球場の消化試合チケットと、「スワローズご声援感謝セール」と名前のすり変わった、去年と変りない安売りタフマン10本組となって無事帰って来たのでした。だー、巨人ファンの方、おめでとうございます。と、広告をお休みをいただいていた間に、日本は大きく変わってしまってたようですね。浦島太郎の思いが手に取るようにわかります。という訳で、関係者・oh!X読者をはじめ多くの皆様、大変ご心配をお掛けしました。ジャストの広告、堂々の「再」復活です。今後とも変わらぬお付き合いをお願いしますね。さて、事務担当からのお願いです。システム事業部関連製品の電話による質問は、弊社営業日（月～金）の13：00～17：00の間にいただけるとスムーズにお話しが進みます。繰り返しになりましたが、是非ともご協力を。

サポート　（有）エヌ・エム・アイ
開発・販売　（株）ジャスト　JAST
〒156　東京都世田谷区宮坂3-10-7 YMTビル3F
Phone.03-3706-9766　FAX.03-3706-9761　BBS.03-3706-7134

超級特価 X68030(CZ-500C)+040turboバンドルセット ¥328,000!

SXパワーアップ委員会とは？

Ver3.1になって、Human環境との融合を見事に果たしたSX-WINDOW。その潜在的なポテンシャルを120%発揮させるべく、FirstClassTechnology内に秘密裏に結成された、それがSXパワーアップ委員会である。

SXパワーアップ委員会シリーズ第1弾は、シャーペンをさらに強化する「シャーペンワープロパック」です。

シャーペンワープロパックをインストールすることによって、シャーペンが限りなくワープロに近い存在へとパワーアップします。

文字の回転や各種タブ、インデントなど、最新ワープロソフトにも負けない表現力を追加するほか、文系ユーザー待望の縦書き表示、縦書きインライン入力もサポート。それでいて、従来通りの軽快さもそのまま継承しています。

日本語に対する親和性がさらに向上したシャーペンは、まさに理想の日本語ワープロと言えるでしょう。

● シャーペンに追加される主な機能
- 縦書き入力
- 文字の回転
- ルーラ(定規)の表示
- 各種タブ(均等割付など)およびインデントの設定*
- 各種禁則処理(追い込み均等など)*
- 行揃えの拡張*
- 段組み印刷

*:パラグラフごとに設定可能

● 動作環境
- SX-WINDOW Ver3.1以上
- 空きメモリ300KB程度

● 付録
- シャーペン外部コマンド開発キット(ライブラリおよびリファレンス)
- IFM ver 4.0

SX-WINDOW用CD-ROM辞書検索ソフト
SX広辞苑《EPWING対応版》

標準価格 ¥19,800　岩波書店「広辞苑第4版」CD-ROM版 バンドルセット ¥43,800

● SX広辞苑《EPWING対応版》の特長
- 豊富でパワフルな検索方法により、必要な情報をすばやくピックアップ。
- 使う側に立って操作系をリニューアル。さらに簡単に、さらに鋭く作業を行なえます。
- 広辞苑の最新版である第4版をもとにしたCD-ROMを使用するので、よりコンテンポラリーなキーワードにアクセス可能です。
- SX-WINDOW上で動作するので記事の参照や引用がとても簡単。シャーペンやEGWordと組み合わせて活用できます。(ただし、広辞苑では大量の引用は禁止されています)
- シャーペンと融合して語句の検索を行なうシャーペン用外部コマンド"LightWing.X"を同梱。複雑な検索を行なう場合はSX広辞苑.Xを、普段よく使う単純な検索にはLightWing.Xを、という使い分けも可能です。
- 広辞苑第4版CD-ROM版と同様に、EPWING(V1)規約にもとづいたCD-ROMタイトルなら、ほとんどのCD-ROMの内容を検索できます。

● 動作環境
- SX-WINDOW 3.0以上
- SX-WINDOW動作中の空きメモリとして1MB以上を推奨
- CD-ROMドライブ(CD-ROM Driver Ver2.0が付属するので、CD-ROM Driverを別途お買い上げいただく必要はありません。CD-ROM Driverのマニュアルや添付ソフト等は付属しません)

X680x0用Ether net接続パック

期間限定特価 ¥78,000

ESP/Xは、Ether netアダプタ「Ether+」と、TCP/IPドライバ、そして基本的なアプリケーションからなるパッケージです。

好評につき、特価期間を延長。2月末日までの御注文まで特価販売です。

- Ether+(米コンパチブルシステムズ社製)
  SCSIインターフェースを介してEther netとX680x0を接続するためのハードウェアです。
  ※10BASE-2対応モデル・10BASE-T対応モデルの2種類があります。
- TCP/IPドライバ
  X680x0でTCP/IPをサポートするドライバ。ソケットも利用可能です。
- 基本的なアプリケーション
  ftp、telnet(いずれもクライアント)等、基本的なアプリケーションを標準添付。ドライバを活用するためのライブラリも付属します。

● 動作環境
- Human68k ver3.0以上
- メモリ常駐量500KB前後
- SCSIインターフェース内蔵機種以外はSCSIボードが必要

コンパクトな筐体のEther+ (14.0×4.0×19.1cm)

※NetWareには未対応です。

68040搭載アクセラレータ
X68040turbo

標準価格 ¥98,000　ヒートシンク別売 ¥1,000

040turboは、68040を搭載したX68030(5インチタイプ)専用のアクセラレータです。040turboを装着することで得られるパフォーマンスは、従来の2〜3倍! 計算、特に浮動小数点演算中心のソフトならば、さらにそれ以上の高速化も望めます。

詳しくはソフトバンク刊「X68040turbo〜A Story of Makeing "After X68030"〜」(BEEPs著)をご覧ください。

040turboは当社のショップBASIC-HOUSEでの直販、および通販でのみお買い求めいただけます。ご注文いただいてから しばらくお待ちいただく場合もありますので、お早めにご注文ください。

SCSI-2対応ドライブ専用ドライバ
CD-ROM Driver Ver2.00
標準価格 ¥4,800

※SCSI-2対応ドライブ以外をお使いの方はバージョンアップの必要はありません。

X680x0用フリーソフトウェア集CD-ROM
FreeSoftwareSelection Vol.2
標準価格 ¥6,000

TAKERUで販売開始!

DoubleBookin'
標準価格 ¥12,800

SX-PhotoGallery
標準価格 ¥15,800

お求めはお近くのパソコンショップ、または当社通販部(TEL:0286-22-9811)へお申し込みください。
通販ご希望の方は、ソフト代金+送料1,000に消費税を加え、ご住所・お名前・電話番号・商品名を明記した紙を同封の上、現金封筒でお申し込みください。

低金利クレジット　通信販売送料 全国一律¥1,000　長期クレジット可能　※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

サポートネット TECOSYS-3 24時間稼動中! (0286)51-1430 (9600bps MNP5)

あの「Personal LINKS」のレンダラー「L/Image」が
ついに移植されました。
その名も……

XL/Image

レンダラー

PHOTO： ©1994 IMAGICA TECHNOSYSTEMS

ワークステーション版にせまる機能を満載して、いよいよ登場！

■いままで、パーソナルユーザーには夢だった高品質・高速の
レイトレーシングによるレンダラーを今、ここにお届けします。
■メタボールのレンダリングができます。もちろん、形状定義も可能。
■コマンドを記述したテキストファイルによって、高画質のレンダリングを
実行できます。また、数値設定により移動、回転、拡大縮小、変形を実行できます。
●80種の各コマンド、数値データのサンプルを提供し、容易にレンダリングが可能。
■光源5種類、光源数最大32個、反射、屈折、透過、影は自由に設定ができます。
●リンクスモデルでは、二重ハイライトをサポート。
●各物体毎に最大32個までの自由な光源設定が可能。
光源種類は、平行・点・円鏡・スポット・環境をサポート。
■テクスチャーマッピングは高品質。もちろん多重マッピングも無制限（メモリ量に依存）。
●マッピング種類は、バンプマッピング、環境マッピングを含む4種類。
8種のソリッドテクスチャーを標準で提供。
■アンチエイリアシングから、デフォーカス・ぼけた影・屈折・モーションブラーが
でき、フォトリアリスティックな映像作成が可能です。
●サンプルを利用して、すぐにレンダリングが可能。
■表現解像度は、自由に設定できます。
●解像度は、コマンドの数値で自由に設定できます。
●65、535色の画面表示に加えて、フルカラー1677万色のファイル出力
をサポート。
■ワイヤーフレーム表示も可能なので、レイアウト確認も容易。
■形状データ、画像データ、サンプルデータ等を含めて80個以上を装備。
■ポリゴンモデラーのデータコンバーターを標準装備しています。
●SUF(Do-GA)形状ファイルコンバーター
●PNA(Z's TRIPHONY DIGITAL CRAFT)形状コンバーター

■動作環境
SHARP X68000シリーズ・X68030シリーズ。Human 68K Ver2.0以上。
4MB以上のRAM搭載。ハードディスク装備のこと。
ハードディスクの空き容量は4MB以上必要。
ディスプレイ条件は512×512の65、535色モードで使えるもの。テキストエディタが必要。
数値演算コプロセッサーの搭載を推奨、未搭載でも可能
5.25インチフロッピードライブ。
ノーマル版、ハイスピード版（X68030＋コプロ用）を同抱。

●XL/Imageはレンダラーです。モデリング、アニメーションソフトは含まれていません。

¥58,000（消費税別）

※シャープX68030シリーズ及びX68000シリーズはシャープ㈱の商品です。
※「Z's TRIPHONY DIGITAL CRAFT」は、㈱ツァイトの商品です。

開発元
株式会社 IMAGICA

販売元
株式会社 IMAGICAテクノシステム
〒141 東京都品川区東五反田2-9-5 サウスウイングビル
TEL.03-5449-3484

T1002179120908 雑誌 02179-12

12月号 1994年12月1日発行（毎月1回1日発行）
Printed in Japan